

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程		
氏 名	小林豪太	学籍番号	1931054
論 文 題 目	微細型 3 コア光ファイバのベクトルモード電磁界解析		
<p>要 旨</p> <p>近年、情報化社会の発展により伝送される通信量は増大の傾向にあり、5G の普及に伴い今後もこの傾向は続くと思われる。これに対応するために通信容量の大容量化が必要とされている。そのために波長多重や多値変調といった技術を用い一本のファイバ当たりの通信容量を増やしてきた。しかし現在流通するシングルモードファイバでは波長多重や多値変調を用いてもこれ以上の大幅な通信容量の増大は困難とみられている。さらなる通信容量の増大のためにシングルモードファイバとは異なる構造の研究が進められている。近年の光ファイバの製造技術の発展に伴い複雑な構造のファイバが作成可能となっている。その中には一本のファイバに複数のコアを微細に配置した微細型マルチコアファイバが存在する。微細型マルチコアファイバにはコア数以上の伝搬モードが存在し、それらを独立し伝送チャネルとして利用でき、伝送容量の増大が期待される。しかし微細型マルチコアファイバの特性解析の研究は少なく、さらなる長距離通信通信の達成には複雑な構造の光ファイバの解析が不可欠である。本研究では微細型マルチコアファイバの構造でもっとも単純な 3 コアファイバについて群論と点整合法を用いた群論適用点整合法でベクトルモード電磁界の解析を行い。さらにベクトルモード電位界の誤差の検証、考察を行った。</p> <p>z 軸方向の電磁界、横方向電界の正規化平均二乗誤差を計算し群論適用点整合法の検証を行い、整合点を 6 点にすることで構造によらず微細型 3 コアファイバの境界条件が精度よく満たされていることがわかった。コアの間隔が広い場合にはコア間の結合が弱くなるため精度が上昇することが明らかになった。</p> <p>シングルモードファイバとの比較を行いコア間隔の増大によりコア間結合が弱くなるため微細型 3 コアファイバの特性はシングルモードファイバに漸近していくことが明らかになった。また電磁界の閉じ込め率の増加でも同様にコア間結合の減少が明らかになった。コア同士が最も接近し隣接している状態では閉じ込め率の高い状態でもコア間結合はコア間隔が広い状態と比べて特別に強く実質的に一つのコアとみなせることがわかった。</p>			

令和2年度修士論文

微細型3コア光ファイバのベクトルモード電磁 界解析

電気通信大学 情報理工学研究科
情報・ネットワーク工学専攻 情報通信工学プログラム
学籍番号 1931054

氏名 小林 豪太

指導教員 來住 直人 教授

副指導教員 松浦 基晴 教授

2021 年 1 月 24 日

概要

近年，インターネット，通信デバイスの普及により通信量は増加の傾向にある．これに対応するため波長多重や多値変調などを用いて伝送量を増大させてきた．しかし現在用いられているシングルモードファイバで波長多重や多値変調でのこれ以上の大幅な通信容量の増大は困難と考えられており，この問題を解決するために一本のファイバに複数のコアを配置する新しい構造のファイバの研究が行われている．その中には一本のファイバに三つのコアを密に配置した微細型3コアファイバが存在する．それらの研究は大容量伝送システムの実験が主であり，詳細な電磁界ベクトルの解析は行われていない．

本研究では微細型マルチコアファイバの構造でも最も単純な3コアファイバを対象とし，群論を適用した点整合法を用いた解析手法でベクトルモード電磁界計算を行い解析手法の誤差の評価，構造が電界に与える影響を考察した

目次

第1章	序論	2
1.1	本研究の背景	2
1.2	本研究の目的と本論文の構成	3
第2章	微細型マルチコアファイバ	4
2.1	微細型マルチコアファイバの構造	4
2.2	群論適用点整合法による固有伝搬モード解析	5
2.3	微細型3コアファイバの伝搬モード	8
2.4	正規化平均二乗誤差を用いた誤差の評価	9
2.5	シングルモードファイバとの比較方法	10
第3章	結果と考察	12
3.1	境界条件の評価	12
3.2	電界のベクトルモード表示と強度表示	19
3.3	PV 特性とコアへの閉じ込め率	23
3.4	シングルモードファイバの電磁界との比較結果	25
第4章	結論	29
	謝辞	30
	参考文献	31
付録A	電磁界計算のための手順式	33
付録B	z 軸方向電磁界, 横方向電界計算プログラムリスト	36

第1章 序論

1.1 本研究の背景

近年，情報化社会の発展により伝送される通信量は増大の傾向にあり，5 G の普及に伴い今後もこの傾向は続くと思われる．これに対応するために通信容量の大容量化が必要とされている [1]．そのために波長多重や多値変調といった技術を用い一本のファイバ当たりの通信容量を増やしてきた．しかし現在流通するシングルモードファイバでは波長多重や多値変調を用いてもこれ以上の大幅な通信容量の増大は困難とみられている．さらなる通信容量の増大のためにシングルモードファイバとは異なる構造の研究が進められている．近年の光ファイバの製造技術の発展に伴い複雑な構造のファイバが作成可能となっている．その中には一本のファイバに複数のコアを微細に配置した微細型マルチコアファイバが存在する．微細型マルチコアファイバにはコア数以上の伝搬モードが存在し，それらを独立し伝送チャネルとして利用でき，伝送容量の増大が期待される．しかし微細型マルチコアファイバの特性解析の研究は少なく，さらなる長距離通信通信の達成には複雑な構造の光ファイバの解析が不可欠である．

1.2 本研究の目的と本論文の構成

本論文では微細型マルチコアファイバの構造でも最も単純な3コアファイバについて群論と点整合法を用いた解析手法でベクトルモード電磁界の解析を行う。さらにベクトルモード電位界の誤差の検証を行うことを目的とする。

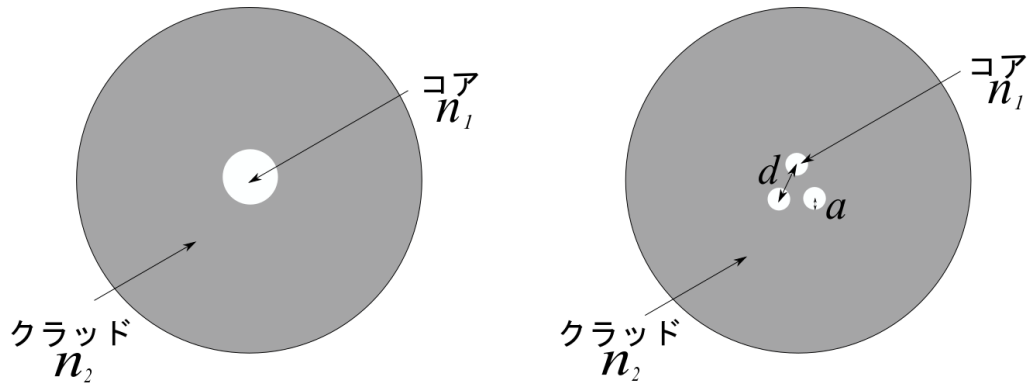
微細型マルチコアファイバはその構造の複雑さゆえに既存のソフトウェアを用いた解析では詳細なが困難であり，伝搬する電界ベクトルの詳細な解析はされていない。さらなる長距離通信の達成のためには複雑な構造のファイバの詳細な解析が必要とされる。

本論文の攻勢について説明する。第2章では微細型3コアファイバの構造とその解析手法について説明し，誤差の評価基準，シングルモードファイバとの比較方法について述べる。第3章では群論と点整合法を用いた解析手法による結果を載せ考察を行う。第4章では本論文の結論を述べる。

第2章 微細型マルチコアファイバ

2.1 微細型マルチコアファイバの構造

図 2.1 に従来のシングルモードファイバ (Single mode fiber:SMF) と微細型 3 コアファイバの断面を示す．一般に光ファイバとはクラッドに石英ガラス，コアに添加物を用いて屈折率を大きくした石英ガラスが使用される．従来用いられるファイバは 2.1(a) のように一つのファイバに一つのコアを配置したものである．本研究で対象とする微細型 3 コアファイバは図 2.1(b) に示すように同一径のコアが 3 つ同心円状に等間隔に配置されたファイバである．ファイバは等方，均質，無損失とした．



(a) 単一コアファイバの断面図 (b) 微細型3コアファイバの断面図

図 2.1: 光ファイバの断面図

微細型マルチコアファイバを用いて信号を伝搬させる場合，コア結合によるクロストークが発生する [2]．これは，クラッド側に向けて指数関数的に減衰するエバネッセント波に起因する．微細型マルチコアファイバではこの結合によりファイバ全体を単一の導波路として伝搬する，複数のモードを用いる．

図 2.2 に微細型 3 コアファイバの断面の構造図を示す．コア半径を a ，コアの中心間距離を d ，とする．さらにコア半径とコア中心間距離の比 d/a を用いる．コア，クラッドの屈折率をそれぞれ n_1, n_2 とし，紙面垂直方向を z 軸とする．各コアに対して図のように極座標 (r_ν, θ_ν) ($\nu = 1, 2, 3$) をとる． z 軸方向に伝搬する伝搬定数 β の光波の電磁界成分 E_z, H_z は式 (2.1) のヘルツホルム方程式を満たす．

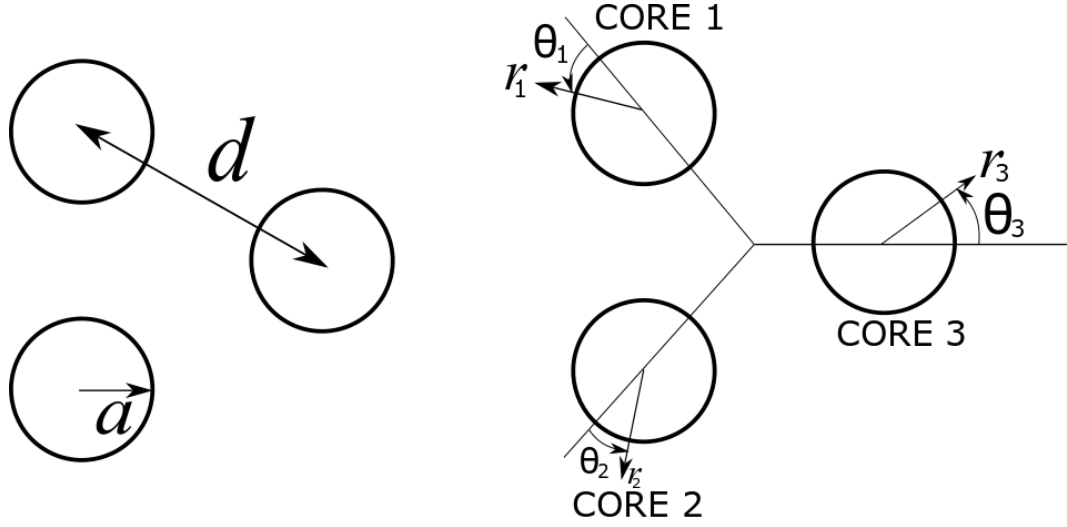


図 2.2: 微細型 3 コアファイバの構造

$$\nabla_t \begin{bmatrix} E_z \\ H_z \end{bmatrix} + (k_i^2 - \beta^2) \begin{bmatrix} E_z \\ H_z \end{bmatrix} = 0 \quad (i = 1, 2) \quad (2.1)$$

$k_i (i = 1, 2)$ はコア ($i=1$), クラッド ($i=2$) の波数, ∇_t は演算子 ∇ の横方向成分である. z 軸に垂直な横方向電磁界成分 E_t, H_t は次の式 (2.2a) のように表現される. μ_0, ϵ_i はそれぞれ真空中の透磁率, コア, クラッドの誘電率となる. \mathbf{a}_z は z 軸方向単位ベクトルとなる.

$$\mathbf{E}_t = \frac{-j\beta}{k_i^2 - \beta^2} (\nabla_t E_z - \frac{\omega\mu_0}{\beta} \mathbf{a}_z \times \nabla_t H_z) \quad (2.2a)$$

$$\mathbf{H}_t = \frac{-j\beta}{k_i^2 - \beta^2} (\nabla_t H_z - \frac{\omega\epsilon_i}{\beta} \mathbf{a}_z \times \nabla_t E_z) \quad (2.2b)$$

2.2 群論適用点整合法による固有伝搬モード解析

電磁界の有限級数展開を用いた解析法の一つに点整合法がある. 点整合法とはコア・クラッドの境界上に有限の離散の整合点を取り, その整合点に境界条件を適用する解析法である. 図 2.3 の各領域は均質・等方・無損失でクラッドの直径は無限に伸ばせるものと仮定する. このような仮定の下では式 (2.1) のヘルツホルム方程式の解の級数展開で表現することが可能となる. 単一コアの場合には次のように表現される [3].

$$E_{z1} = \sum_m a_m E_{1m} \quad (2.3a)$$

$$H_{z1} = \sum_m b_m H_{1m} \quad (2.3b)$$

$$E_{z2} = \sum_m c_m E_{2m} \quad (2.3c)$$

$$H_{z2} = \sum_m d_m H_{2m} \quad (2.3d)$$

E_{1m}, H_{1m} はコアにおける式 (2.1) の基本解で, E_{2m}, H_{2m} はクラッドにおける式 (2.1) の基本解である. a_m, b_m, c_m, d_m はそれぞれ展開係数 (未知係数) である. これらを用いて横方向電界を表し電磁界の境界条件を適用する. 境界条件は z 軸方向の電磁界と横断面電磁界 (2.2 式) のコアクラッド境界の接線方向成分である. しかし境界上で連続的に境界条件を適用させ満足させることは困難なため各整合点で境界条件を適用する. 適用することで同時連立方程式が導かれ, 非零解を持つ条件から周波数と伝搬定数の関係である分散特性がわかり, その条件下で係数行列の階数を一つ減じた同時方程式を解くことにより電磁界計算のために必要な未知係数が計算される. 整合点を増やすことで高精度に未知係数を計算できるが同時に計算量も増加する. 境界条件は電磁界それぞれの接線方向成分の連続性であるため一つの点につき条件は 4 個となる.

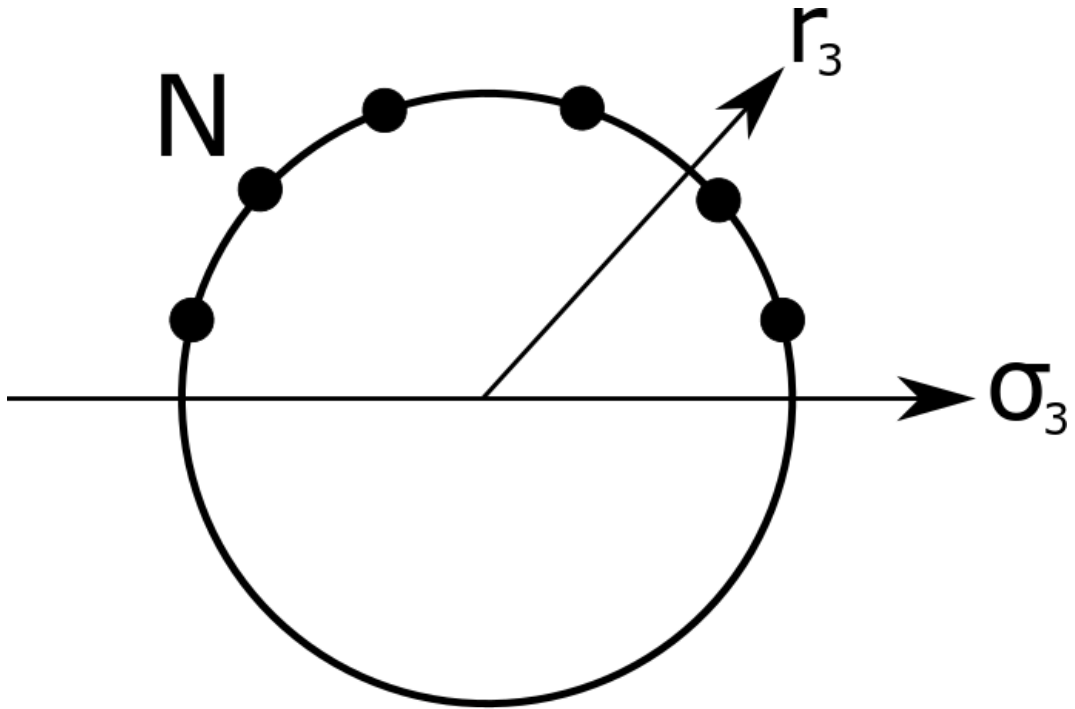


図 2.3: 離散点の画像

図 2.2 のような多重円柱座標において, r_k, θ_k を $k(k=1,2,3)$ 番目のコアを中心とする座標とすると, コア内では

$$E_z = \sum_m (a_{em}^k \cos m\theta_k + a_{om}^k \sin m\theta_k) J_m(\chi_1 r_k) \quad (2.4a)$$

$$H_z = \sum_m (b_{em}^k \cos m\theta_k + b_{om}^k \sin m\theta_k) J_m(\chi_1 r_k) \quad (2.4b)$$

$$\chi_1 = \sqrt{k_1^2 - \beta^2} \quad (2.4c)$$

のように表現し、クラッド内ではつぎのように表現する.

$$E_z = \sum_{k=1}^n \sum_m (c_{em}^k \cos m\theta_k + c_{om}^k \sin m\theta_k) K_m(\chi_2 r_k) \quad (2.5a)$$

$$H_z = \sum_{k=1}^n \sum_m (d_{em}^k \cos m\theta_k + d_{om}^k \sin m\theta_k) K_m(\chi_2 r_k) \quad (2.5b)$$

$$\chi_2 = \sqrt{k_2^2 - \beta^2} \quad (2.5c)$$

ここで $J_m(x), K_m(x)$ はそれぞれ第一種ベッセル関数と第二種変形ベッセル関数である. クラッド領域ではすべてのコアからの寄与を重ね合わせることでコア間の結合を含めた電磁界を表現する. これらの展開係数はコア・クラッドの境界における境界条件を適用することで計算可能となる. しかしコアが多重な場合伝搬定数が縮退したモードが存在するためモードを分類して解析することが難しくなる. さらに本研究で対象とする微細型3コア光ファイバの場合, 仮に一つのコア当たり10点の整合点をとると未知係数は120となり, 計算時間の増大と計算精度の低下を招く. これを回避するため群論を導入する. 群論 [5] を導入することで微細型マルチコアファイバの幾何学的対称性による電磁界分布の対称性を利用することが可能となり, 計算精度の向上が可能となる.

正 n 角形の全頂点に同一のコアが配置されている構造は正 n 角形と同一の幾何学対称性を保有している. そのような対称性を不変に保つ回転や軸に対する折り返しの「対称操作」の集合は数学の群の性質を満たしている. 正 n 角形に対する対称操作は, 図形の中心を原点とする角度 $2\pi/n$ の回転と, n 本の対称軸 $\sigma_1, \sigma_2, \dots, \sigma_n$ に対する折り返しの計 $2n$ 個である. このような対称形状のもとでの解は整数 l を用いて記述される. 微細型 n コアファイバでは正三角形の中心と n 番目のコアの中心を結ぶ軸 σ_n に関する偶関数と奇関数に分類される. 結果としてコアでの z 軸方向電磁界については

$$E_z = \sum_m (P_{ek}^{(l)} a_{em} \cos m\theta_k + P_{ok}^{(l)} a_{om} \sin m\theta_k) J_m(\chi_1 r_k) \quad (2.6a)$$

$$H_z = \sum_m (P_{ek}^{(l)} b_{em} \cos m\theta_k - P_{ok}^{(l)} b_{om} \sin m\theta_k) J_m(\chi_1 r_k) \quad (2.6b)$$

となり, クラッド内では

$$E_z = \sum_{k=1}^n \sum_m (P_{ek}^{(l)} c_{em} \cos m\theta_k + P_{ok}^{(l)} c_{om} \sin m\theta_k) K_m(\chi_2 r_k) \quad (2.7a)$$

$$H_z = \sum_{k=1}^n \sum_m (P_{ek}^{(l)} d_{em} \cos m\theta_k - P_{ok}^{(l)} d_{om} \sin m\theta_k) K_m(\chi_2 r_k) \quad (2.7b)$$

係数 $P_{ek}^{(l)}, P_{ok}^{(l)}$ は

$$P_{ek}^{(l)} = \begin{cases} \cos \frac{2\pi kl}{n} & (E_z \text{ が偶関数}) \\ \sin \frac{2\pi kl}{n} & (E_z \text{ が奇関数}) \end{cases} \quad (2.8a)$$

$$P_{ok}^{(l)} = \begin{cases} -\sin \frac{2\pi kl}{n} & (E_z \text{ が偶関数}) \\ \cos \frac{2\pi kl}{n} & (E_z \text{ が奇関数}) \end{cases} \quad (2.8b)$$

となる [3][4]. 整数 l は固有モードの対称形状と縮退関係を表す. 具体的には $l = 0, [n/2]$ の時には偶モードと奇モードは非縮退であり, それ以外では縮退である.

式 (2.6),(2.7) では各コアの展開係数が共通の組であるため式 (2.4),(2.5) と比較し展開係数を少なく, 幾何学対称性により全体の電磁界を構造の一部で表現し, 境界条件を適用する領域も削減可能となる. 群論を適用することにより固有モードを分類し高効率で解析が可能となる. 多重円柱座標における電磁界の計算手順については付録 A に示す.

2.3 微細型3コアファイバの伝搬モード

微細型マルチコアファイバは複数の導波路が平行に配置され結合しているため, コア数以上の伝搬モードが存在する. このような導波路の集まりは一本の導波路とみなすことが可能で, この時の伝搬モードは結合導波路の固有モードといわれる.

微細型3コアファイバでは近似会による解析により図 2.4 に示される6つのモードを有する [6]. これらのモードは独立した伝送チャネルとして利用が可能なためシングルモードファイバと比べて大きな伝送容量を有する.

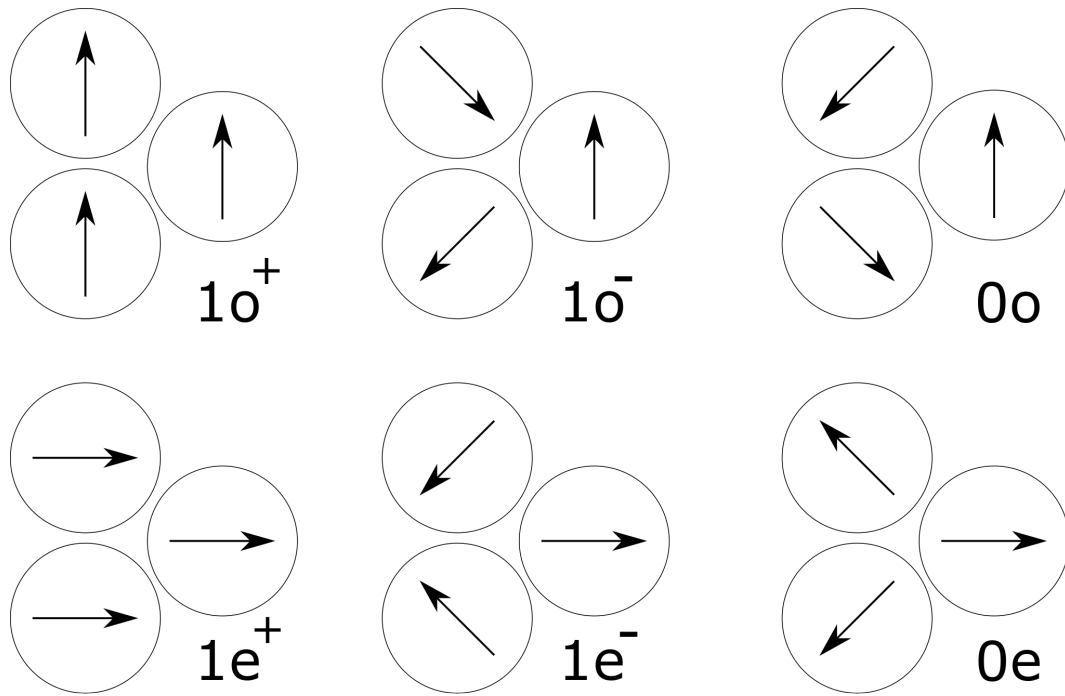


図 2.4: 微細型3コアファイバの基本伝搬モード

モードの分類は縮退関係を表す整数 l と z 軸方向電磁界の偶関数, 奇関数を表す e, o で分類される. モード $1o^+, 1e^+$, モード $1o^-, 1e^-$ はそれぞれ2重縮退する.

2.4 正規化平均二乗誤差を用いた誤差の評価

点整合法では境界上に離散的に取られた整合点上のみにおいて境界条件が満たされており，それ以外の境界における境界条件の成立は担保されていない．そこで境界上における電磁界の接線方向成分のコア側，クラッド側の値を用いて，次の式 (2.9) で定義する正規化平均二乗誤差 (normalized mean-square error: NMSE) により精度を評価する．

$$NMSE = \frac{\int_L (\phi_{core} - \phi_{clad})^2 ds}{\int_L \phi_{core}^2 ds} \quad (2.9)$$

ϕ_{core}, ϕ_{clad} はそれぞれ境界上の電磁界の接線成分のコア，クラッド側の値で， $E_z, H_z, E_\theta, H_\theta$ のいずれかとなる．積分は図 2.3 の 3 番目のコアの境界 L 上の周回線積分である．各 d/a でこの数値を計算し精度を評価した．

解析で用いられるパラメータとして，コアの本数 n は 3 とし，一つのコアの半径 a と隣接するコアの中心間の距離の比 d/a を用いる．さらにコアクラッドの比屈折率差，正規化周波数 V と正規化伝搬定数 P を用いる．

$$V = k_0 a \sqrt{n_1^2 - n_2^2} \quad (2.10)$$

$$P = \frac{(\beta/k_0)^2 - n_2^2}{n_1^2 - n_2^2} \quad (2.11)$$

次に光ファイバ電磁界のコアへの閉じ込め率について検討する．光ファイバ電磁界のコアへの閉じ込め率はコアとクラッドの全領域を含めた断面全体の伝搬光パワに対するコアを伝搬する伝搬光パワの比とする．伝搬光パワ P_g はポインティングベクトルを用いて z 軸を進行方向とすると次のようになる [7]．

$$P_g = \frac{1}{2} \int_0^{2\pi} \int_0^\infty (E_x H_y^* - E_y H_x^*) r dr d\theta \quad (2.12)$$

半径が a のシングルモードファイバについてファイバ中心を原点とする極座標 (r, θ) をとる．従来用いられる光ファイバは比屈折率差 $\Delta \ll 1$ なので弱導波近似を行うことが可能となる．そのような状況下では y 方向に偏光したモード $LP_{\nu\mu}$ を考えると横方向電磁界 E_x, E_y, H_x, H_y は次のようになる．

$$E_y = \begin{cases} A_1 J_\nu(ur/a) \\ A_2 K_\nu(wr/a) \end{cases} \cos(\nu\theta) \quad \begin{matrix} : 0 \leq r \leq a \\ : r \geq a \end{matrix} \quad (2.13a)$$

$$H_x = -Y E_y, \quad E_x = H_y = 0 \quad (2.13b)$$

$$u = [(n_1 k_0)^2 - \beta^2]^2 a \quad (2.13c)$$

$$w = [\beta^2 - (n_2 k_0)^2]^2 a \quad (2.13d)$$

A_1, A_2 は境界条件から決定される電界振幅係数， Y は特性アドミタンスである． u, w はそれぞれコア，クラッドにおける横方向規格化伝搬定数である．これを式 (2.12) に代入することで弱導波近似の下でのコア内パワー P_{co} と P_g の比

$$\Gamma(V) = \frac{P_{co}}{P_g} = \left(\frac{w}{V}\right)^2 \left[1 - \frac{J_\nu^2(u)}{J_{\nu+1}(u)J_{\nu-1}(u)}\right] \quad (2.14)$$

から、コアへの電磁界の閉じ込め率 $\Gamma(V)$ を定義する.

次に式 (2.10),(2.13c) より $P = w^2/V^2$ となり, 参考文献 [9] より

$$\begin{aligned}\frac{d(VP)}{dV} &= 2\frac{w}{V}\frac{dw}{dV} - \frac{w^2}{V^2} \\ &= P \left[1 - \frac{2J_\nu^2(u)}{J_{\nu+1}(u)J_{\nu-1}(u)} \right]\end{aligned}\quad (2.15a)$$

となる. これを用いると, 最終的にコア中を伝搬する光パワ P_{co} とクラッドも含めた断面全体の伝搬光パワ P_g の比 $\Gamma(V)$ は

$$\begin{aligned}\Gamma(V) &= \left(\frac{w}{V}\right)^2 \left[1 - \frac{J_\nu^2(u)}{J_{\nu+1}(u)J_{\nu-1}(u)} \right] \\ &= \frac{1}{2}P \left[1 + 1 - \frac{2J_\nu^2(u)}{J_{\nu+1}(u)J_{\nu-1}(u)} \right] \\ &= \frac{1}{2}P + \frac{1}{2}\frac{d(VP)}{dV} \\ &= P + \frac{1}{2}V\frac{dP}{dV}\end{aligned}\quad (2.16)$$

となる.

2.5 シングルモードファイバとの比較方法

微細型マルチコアファイバは各コアの結合, 干渉によりシングルモードファイバとは異なった特性を持つ. しかし d/a を大きくすることで各コアを相対的に離れた場合シングルモードファイバに似た電磁界分布になると考えられる. どの程度 SMF に近い電磁界分布を持つのかについて微細型マルチコアファイバと SMF の横方向電界の NMSE を計算した. ここでの NMSE は

$$NMSE = \frac{\int_0^{2\pi} \int_0^R |E_{multi} - \alpha E_{single}|^2 r d\theta dr}{\int_0^{2\pi} \int_0^R |E_{single}|^2 r d\theta dr}\quad (2.17)$$

とする. E_{single}, E_{multi} はそれぞれ SMF, 微細型マルチコアファイバの横方向電界である. α を変化させ NMSE が最小となる値を計算することで二つの電磁界分布が最も似ている状態を取得した. SMF と微細型 3 コアファイバの比較として次の 2 パターンでの比較を行った.

(a) SMF と微細型 3 コアファイバのコア中心を合わせた場合

(b) SMF と微細型 3 コアファイバのファイバ中心を合わせた場合

(a) では微細型 3 コアファイバの一つのコアと SMF のコアの中心を図 2.5 のように重ねて電磁界を比較し, NMSE を計算する. 比較領域はコアの中心から $d/2$ を半径とする円内部とする. SMF のパラメータは微細型 3 コアファイバと同一の V, Δ, a, λ とする. これにより個々のコアがどの程度 SMF に似た電界分布を持っているか計算する.

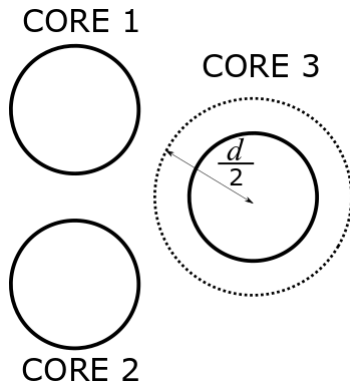


図 2.5: (a) コアの中心を重ねた場合の概念図

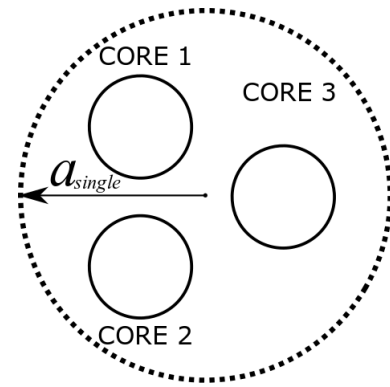


図 2.6: (b) ファイバの中心を重ねた場合の概念図

次に (b) では SMF で用いられる特性の一つであるモードフィールド径 (mode field diameter: MFD) [7] について SMF と微細型マルチコアファイバの比較を行う。従来用いられている SMF ではコア半径が $10\mu m$ 程度と小さく、また光がクラッドにも広がっているためコア・クラッド境界が判別しにくい。そのため光ファイバ断面内の光強度が最大値の $1/e^2$ (e : 自然対数の底) になる直径をモードフィールド径と呼んでいる。微細型マルチコアファイバの場合コア同士が結合し実質的に一つのコアとみなすことが可能であり、最も微細型マルチコアファイバに近い電界分布を持つ SMF の MFD を求める。

実際には微細型 3 コアファイバ全体を一つのコアとみなし電磁界を計算し最も類似性が高い SMF の MFD を求める。まず図 2.7 のように微細型 3 コアファイバと半径 a の SMF を重ね NMSE を計算する。その後 SMF のコア径 a を a' に変化させ再度 NMSE を計算する。図 2.7 ではコア半径を大きくしているが場合によってはコア半径を小さくしている。これを繰り返し NMSE が最少となる電磁界をもつ SMF のコア径を探索する。NMSE の最小値の探索には滑降シンプレックス法を用いた [8]

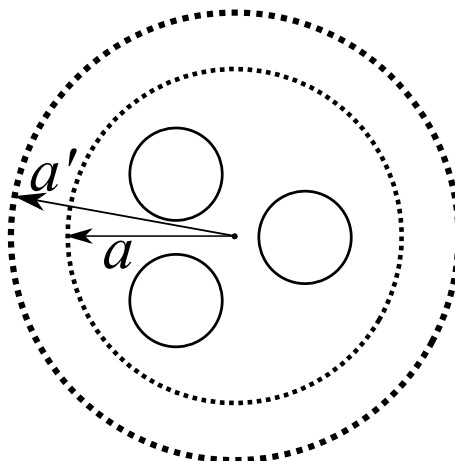


図 2.7: 微細型 3 コアファイと SMF をコア半径を変えて比較する図

第3章 結果と考察

3.1 境界条件の評価

図 3.1, 3.2 に $d/a = 2$ の微細型 3 コアファイバの 6 個のモードの横方向電界分布図をベクトルで表示する．ファイバのパラメータは $\Delta = 0.03, V = 2, d/a = 2$ (コア同士が接触) である．各コアの基本モードが結合した固有伝播モードである．モードは整数 l と対称性で分類され偶モードを e ，奇モードを o として表現する．3 コアでは $l = 0, 1$ となる．

図 3.1, 3.2 は近似解で求めた解法による図 2.4 の電界分布とよく一致している．ただモード $1e^+, 1o^+$ 以外では各コアの結合の影響でコア内の電界分布は近似解の電界分布と比べて一様に同じ方向を示してはいないことがわかる．このベクトルの違いについては 3.4 節にて考察を行う．比較のために図 ??, 3.4 に $d/a = 3$ の微細型 3 コアファイバの各モードの横方向電界分布を示す． $d/a = 3$ ではコア内の電界分布は $d/a = 2$ と比べて図 2.4 に一致している事が確認できる．

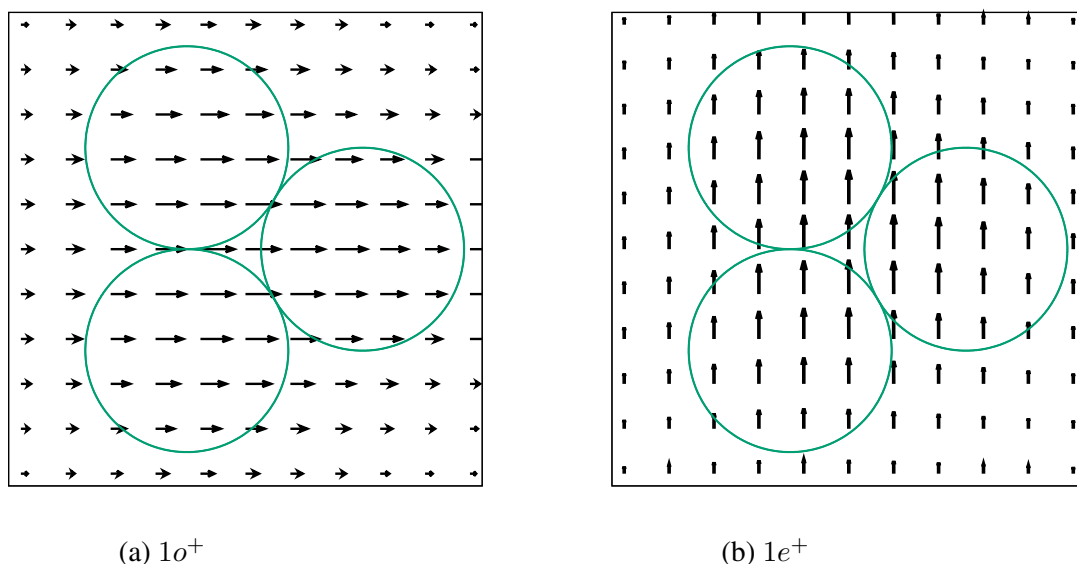
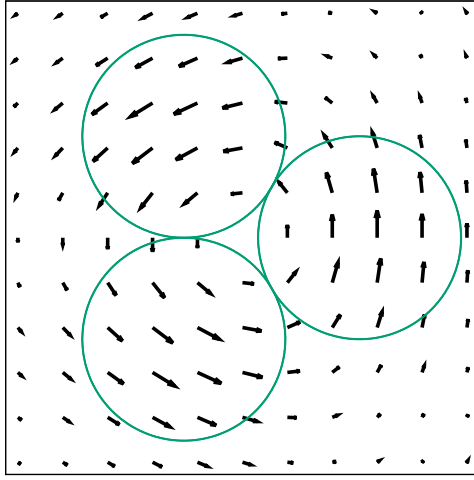
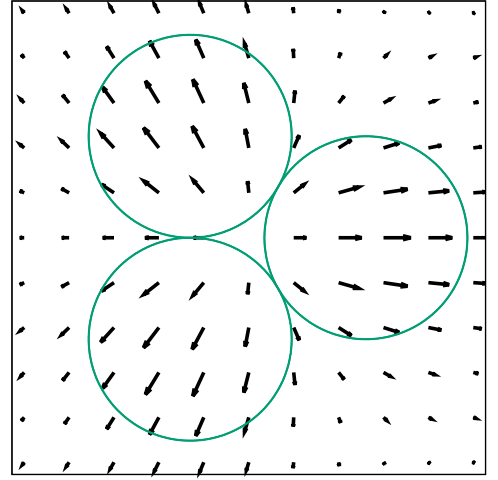


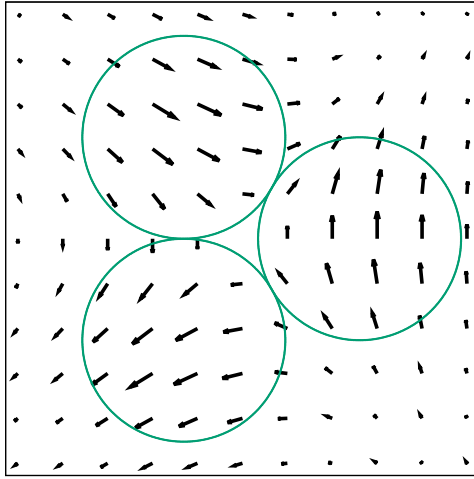
図 3.1: $d/a = 2$ におけるモード $1o^+, 1e^+$ の横断面電界



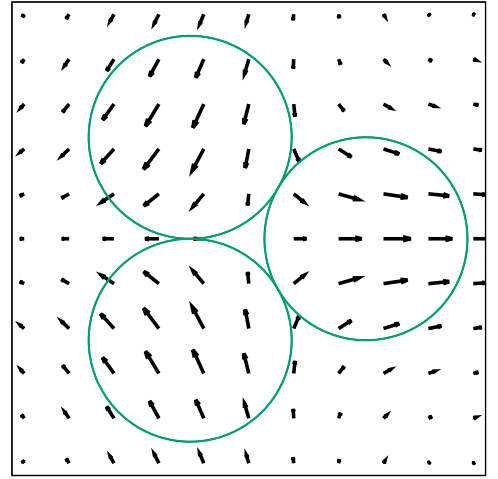
(c) $0e$



(d) $0o^+$

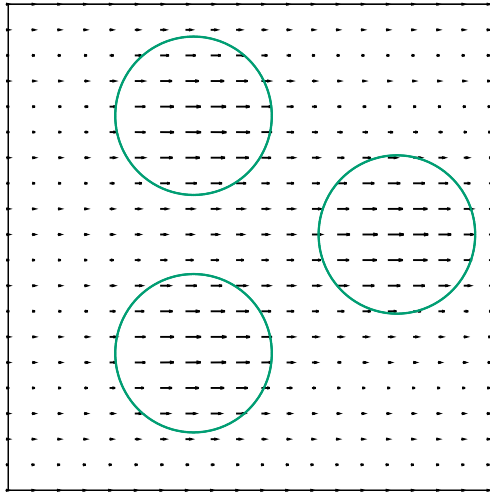


(b) $1o^-$

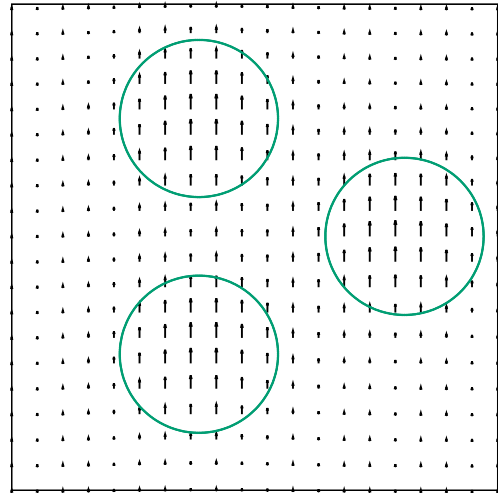


(b) $1e^-$

図 3.2: $d/a = 2$ におけるモード $0e, 0o, 1o^-, 1e^+$ の横断面電界

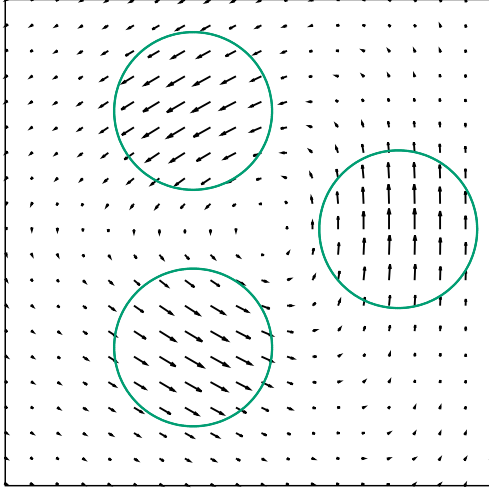


(a) $1o^+$

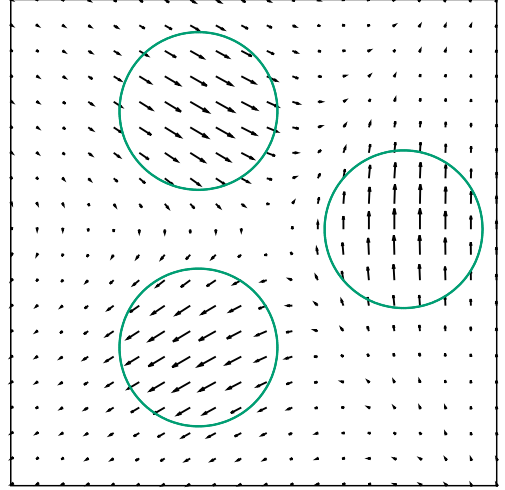


(b) $1e^+$

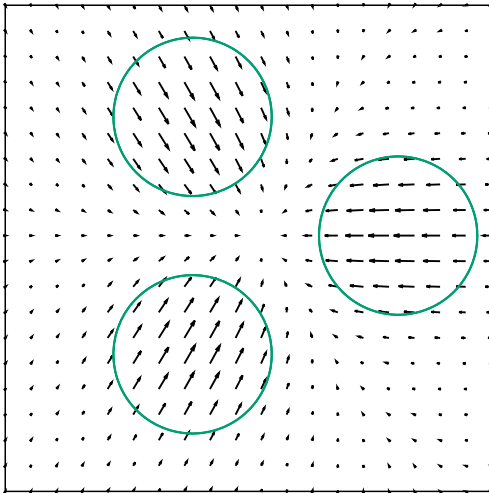
図 3.3: $d/a = 3$ におけるモード $1o^+, 1e^+$ の横断面電界



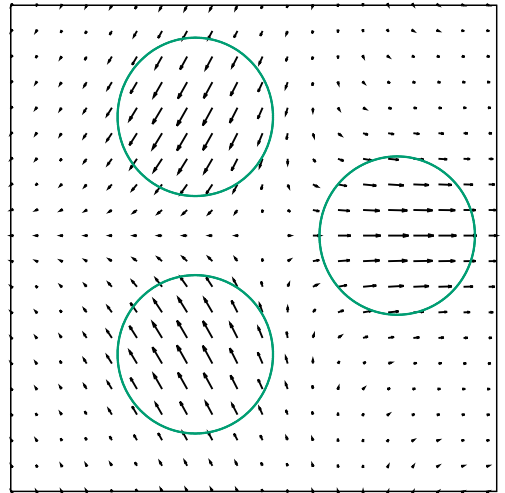
(c) $0e$



(d) $0o^+$



(b) $1o^-$



(b) $1e^-$

図 3.4: $d/a = 3$ におけるモード $0e, 0o, 1o^-, 1e^+$ の横断面電界

次に電磁界の境界条件の精度を検証する．まず初めに式 (2.6),(2.7) における整合点の数について検証を行う．コア半周に取られた整合点の数を N とし， N を変化させモード $1e^+$ の z 軸方向電界磁界，横方向電界の NMSE を計算し図 3.5 に示す．パラメータは $\Delta = 0.003, d/a = 2, V = 1$ である．NMSE は core3 のコア・クラッド境界上で計算した．これにより整合点の数が精度に与える影響を検証する．さらに図 3.6 にコア・クラッド境界上でのコアの電磁界とクラッドの電磁界の差を示す．なお電界と磁界の次元を統一するために z 軸方向磁界 H_z にはクラッドにおけるインピーダンス Z_c を乗算している．

図 3.5 では N の増加に伴い z 軸方向電磁界，横方向電界共に減少が確認された． $N = 2$ の NMSE が最大となる場合で NMSE が $10^{-1} \sim 10^{-2}$ 程度で N の増加に伴い減少している．整合点の増加に伴いより計算量は増加するが境界条件を満たす点が増えたことにより正確に電磁界を計算可能となった．本研究では特に指定がない場合は $N = 6$ として計算を行っている．

図 3.6 のコア・クラッド境界での電磁界の誤差から実際に整合点では誤差が 0 となり境界条件が満たされ，他の領域では誤差が増加減少していることが確認された．この誤差は整合点の数 N の増加により低減することが可能である．

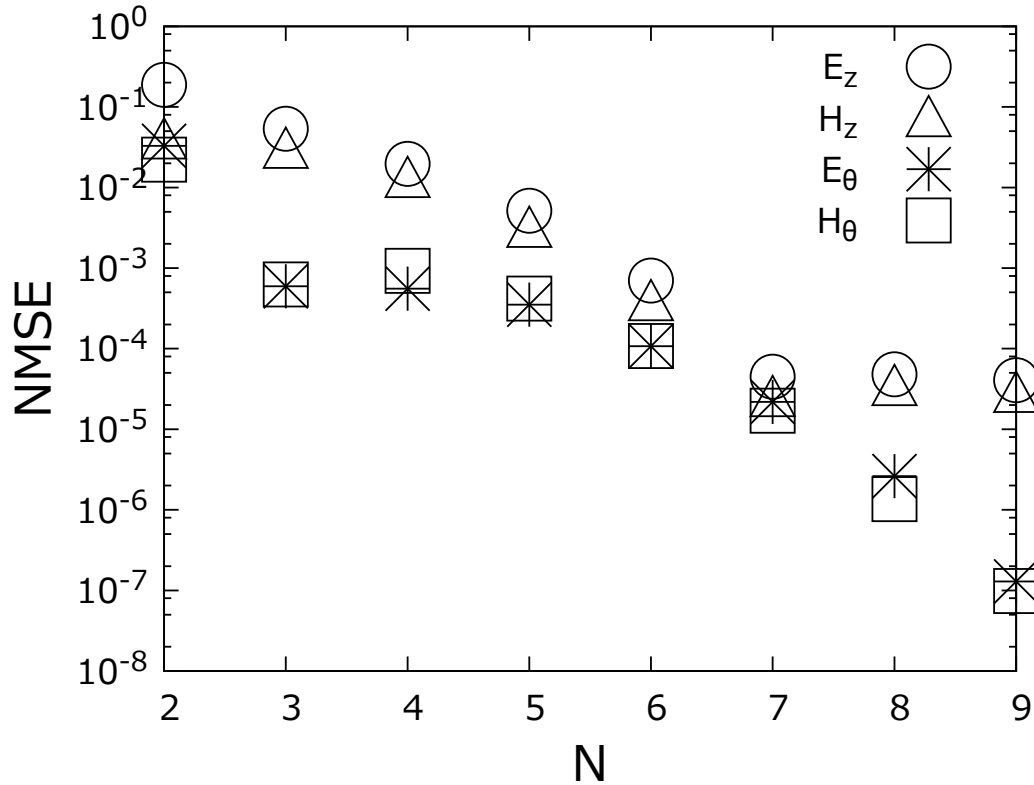


図 3.5: 整合点の数に対する誤差

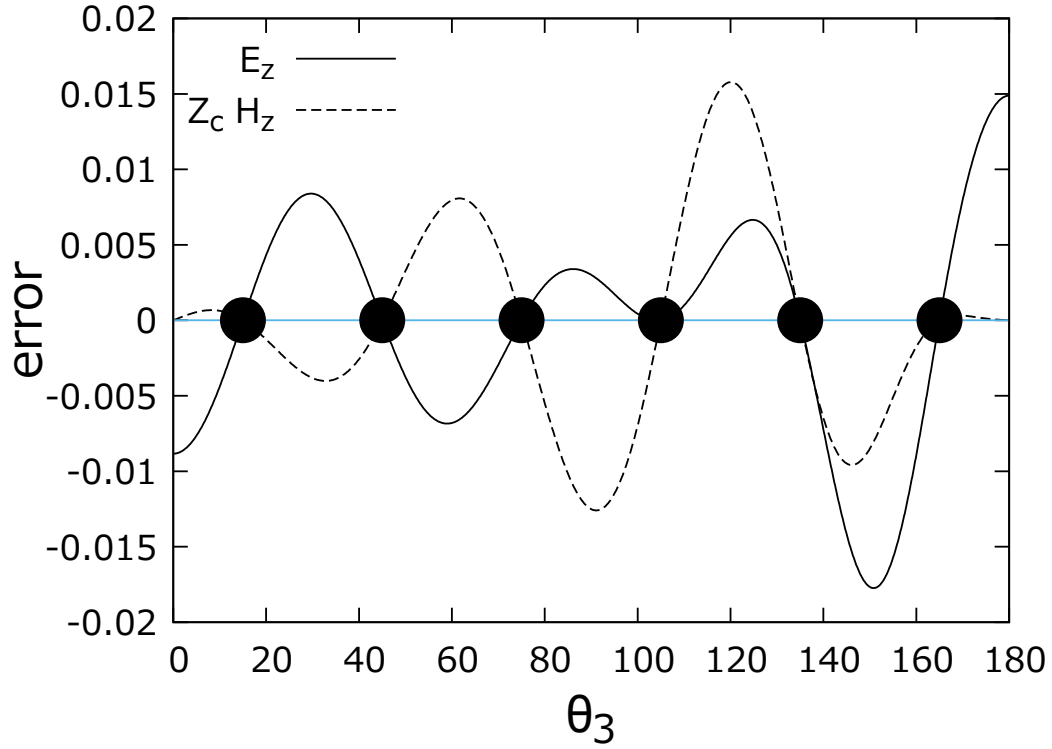


図 3.6: 境界上の電磁界の誤差

図 3.7, 3.8 に各 d/a における z 軸方向電磁界, 横方向電界の NMSE のまとめを示す. パラメータはモード $1e^+$ では $\Delta = 0.003, V = 1.0$ でモード $1e^-$ では $\Delta = 0.003, V = 2.0$ である. コア半径 a とコア中心間距離 d を変化させ core3 の境界上で NMSE を計算した. 図 3.7, 3.8 両方の図で d/a の増加に伴い NMSE は減少することが確認された. d/a は相対的なコアの距離を表している. d/a が大きい場合にはコア同士が相対的に離れていることになるためコア同士の結合が弱く少ない展開項数でも電磁界がより正確に計算されるためと考える. 逆に $d/a = 2$ ではコア同士が最も接近し接触している状態でありコア同士の結合は最も強くなり同じ展開項数では精度が悪くなると考える. $d/a = 2$ で最も精度が悪い場合でも NMSE は 10^{-3} 程度でありコアとクラッドは 0.1% 程度の精度で計算が出来ている.

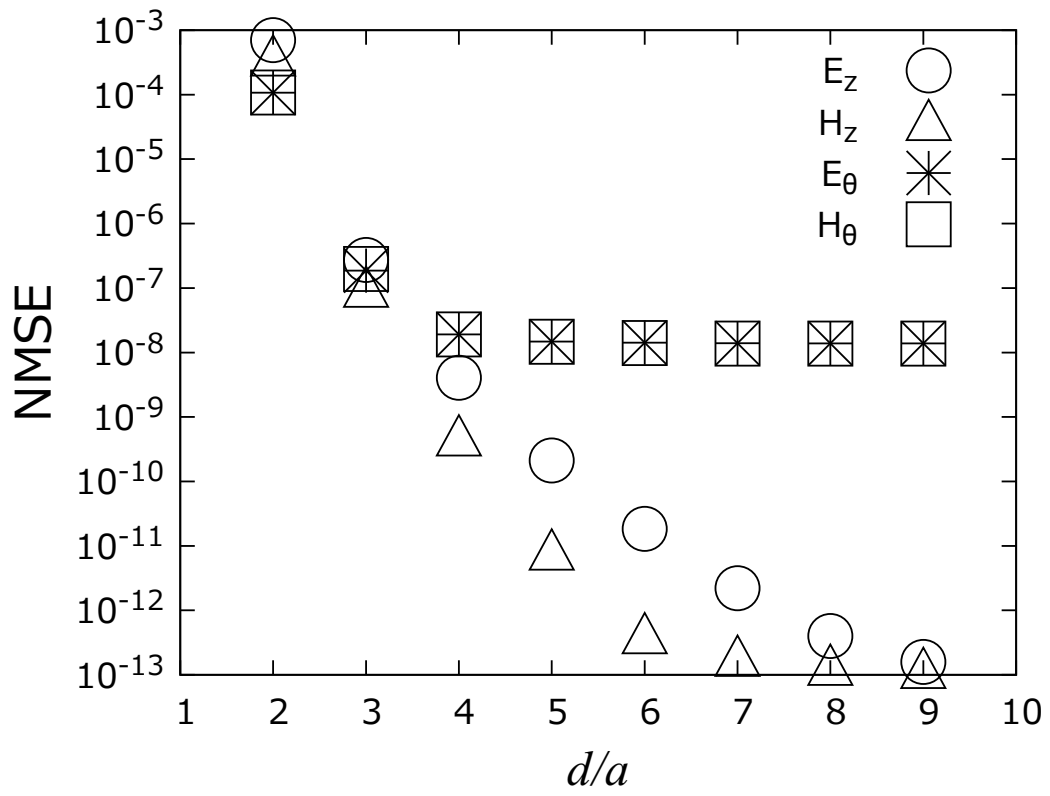


図 3.7: モード $1e^+$ に対する NMSE の d/a 依存

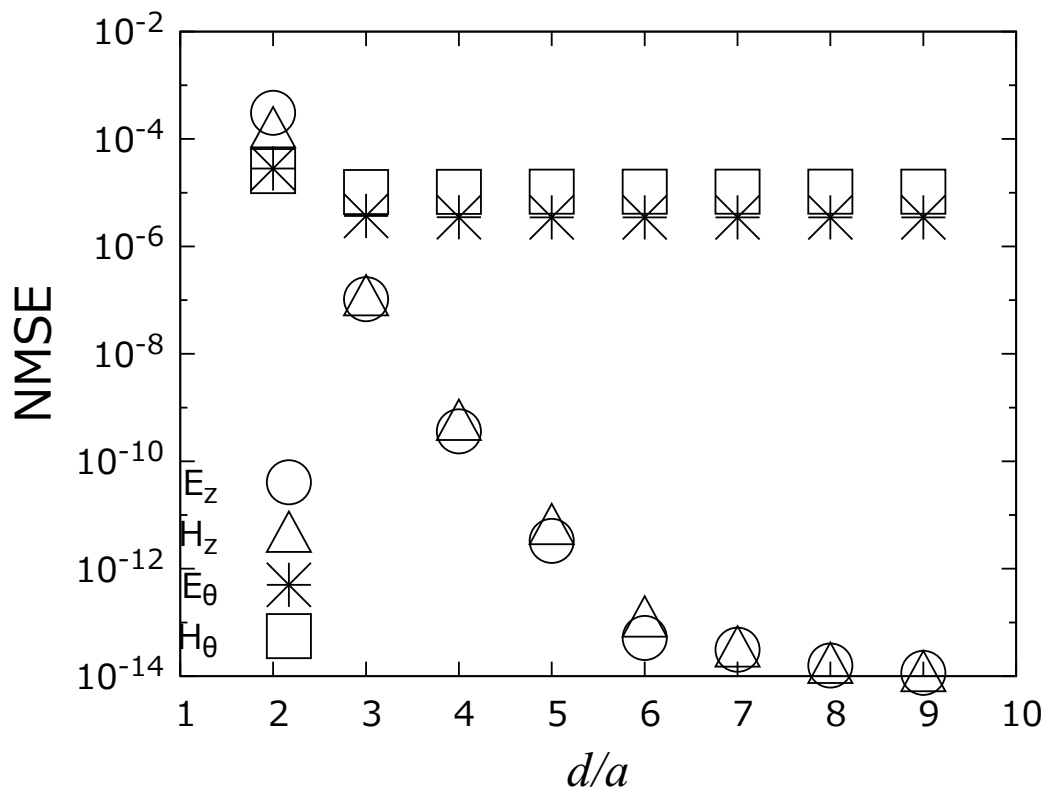


図 3.8: モード $1e^-$ に対する NMSE の d/a 依存

図 3.9 に正規化周波数 V を変化させた場合の z 軸方向電磁界, 横方向電界の NMSE のまとめを示す. パラメータは $d/a = 2, \Delta = 0.003$ モードはモード $1e^+$ ある. 図 3.9 より V の増加に伴い NMSE の増加が確認された. V が小さい場合には正規化伝搬定数 P も小さい値をとるため電磁界のコアへの閉じ込めは弱く, 電磁界はファイバに平面上に広がるため電磁界表現は容易となるため誤差は小さくなる. 逆に V の増加に伴いコアへの閉じ込め率が高くなり誤差は増大したと考える. 計算精度は最も悪い場合で NMSE は 10^{-2} 程度となった. 以上の解析結果より微細型 3 コアファイバにおいて群論を適用した点整合法を用いて精度よく電磁界を計算可能なことが確認された.

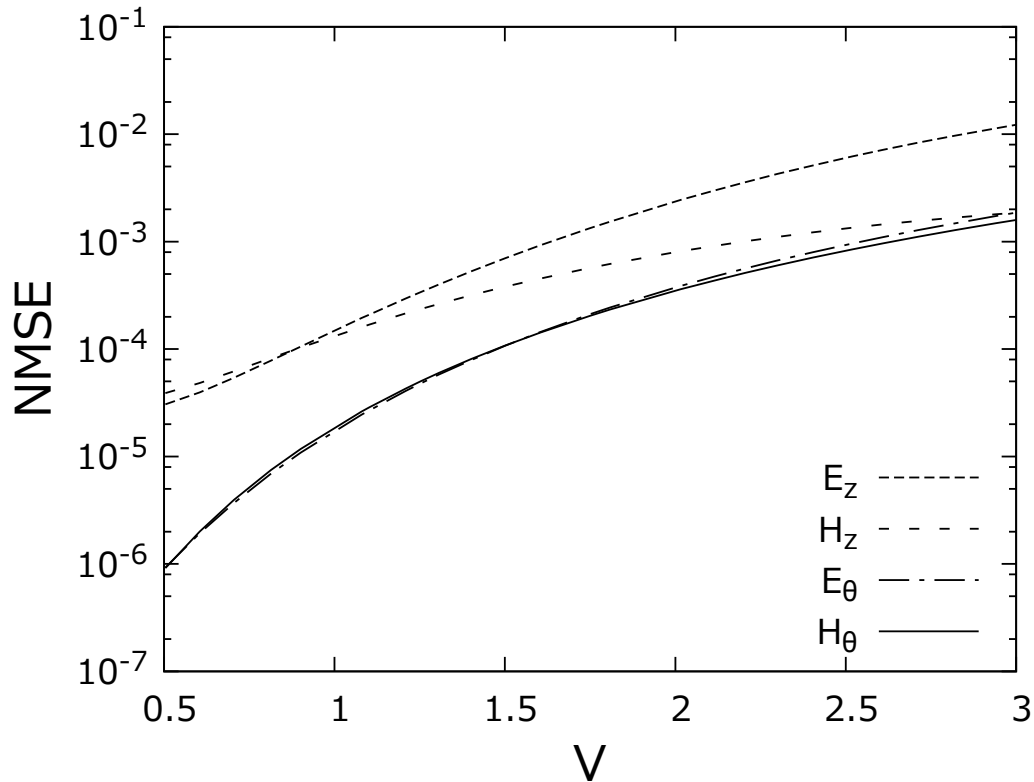
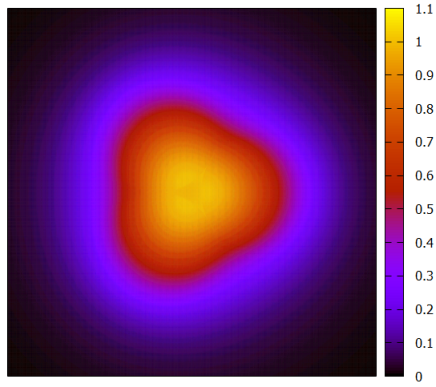


図 3.9: モード $1e^+$ に対する NMSE の V 依存

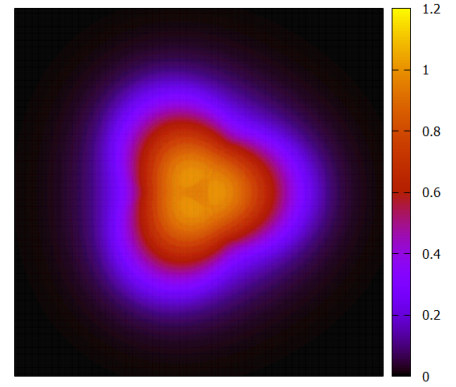
3.2 電界のベクトルモード表示と強度表示

$d/a = 2, 3, 4$ においてモード $1e^+$ についてベクトル分布から強度分布を作成し図 3.10～3.12 に示す. パラメータは $\Delta = 0.003$ で正規化周波数 V を 1, 1.5, 2.0 と変化させた. シングルモードファイバの基本モードでは強度分布の最大値はコアの中心となるが微細型 3 コアファイバでは各コアの影響により強度分布の最大値は各コアの中心にならずファイバの中央に寄っていることが確認できた. 特に図 3.10 では $d/a = 2$ (コアが最接近し接触している) ためコア同士の結合が強く強度分布では各コアの区別がつかなくなっている. d/a の増加に伴い各コアが相対的に離れるため結合は弱くなり強度分布での各コアがはっきりと確認できるようになった. コアが相対的に近い場合, V が小さい場合にはコア同士

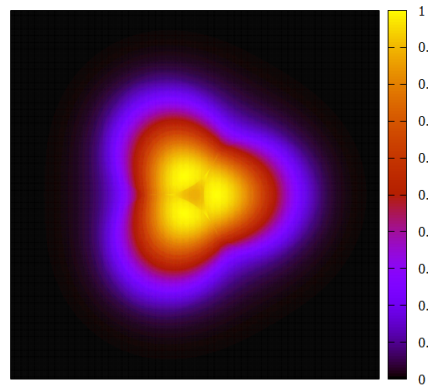
の結合が強く実質的に一つのコアと見せると考えられる.



(a) $V = 1.0$

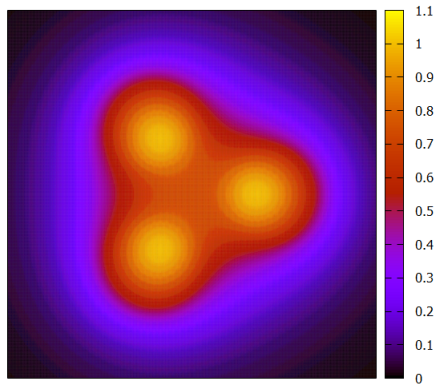


(b) $V = 1.5$

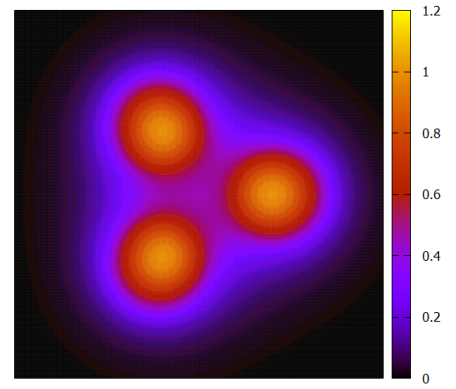


(c) $V = 2.0$

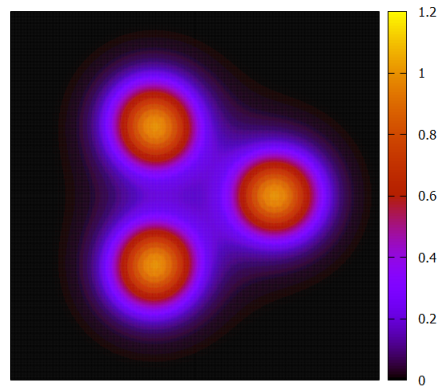
図 3.10: $d/a = 2$ のモード $1e^+$ の横方向電界強度分布



(a) $V = 1.0$

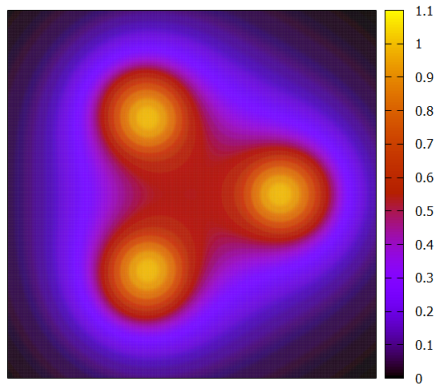


(b) $V = 1.5$

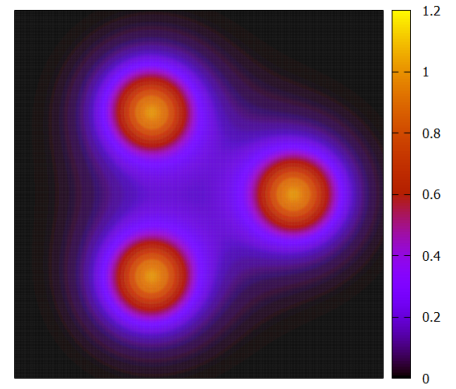


(c) $V = 2.0$

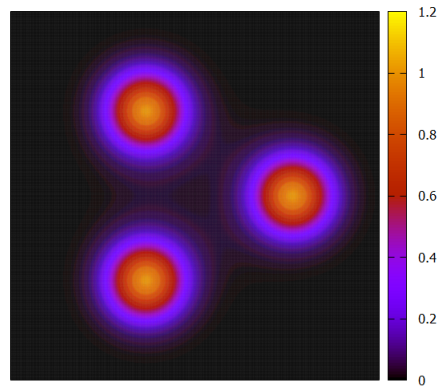
図 3.11: $d/a = 3$ のモード $1e^+$ の横方向電界強度分布



(a) $V = 1.0$



(b) $V = 1.5$



(c) $V = 2.0$

図 3.12: $d/a = 4$ のモード $1e^+$ の横方向電界強度分布

3.3 PV 特性とコアへの閉じ込め率

図 3.13, 3.14 に $\Delta = 0.003, d/a = 2, 3, 4$ とシングルモードファイバでのモード $1e^+$ とモード $1e^-$ の正規化伝搬定数 P と正規化周波数 V の関係を示す。次に式 (2.16) に定義した電界のコアへの閉じ込め率を図 3.15 に示す。図 3.13 では $d/a = 2$ がシングルモードファイバと比べて最も異なる VP 特性を持ち、 d/a の増加に伴い VP 特性はシングルモードファイバに似た特性となる。これは $d/a = 2$ の増加に伴いコア間の結合は弱くなるため、各コアがシングルモードファイバの場合に漸近していくためと考えられる。また光ファイバは式 (2.11) より導波モードでは $n_2 \leq \beta/k_0 \leq n_1$ を満たしており、正規化伝搬定数 P は β/k_0 が n_2 から n_1 の間のどのあたりに相当するかを示している。このことから微細型 3 コアファイバはシングルモードファイバと同一の正規化周波数 V で比べた際によりコアに近い特性と持つと考えられる。これは微細型 3 コアファイバの場合では一つのコアからクラッドへしみだした光の振幅が他のコア付近でも大きいため、その影響が表れていると考える。強度分布の図 3.10 からファイバの中心やコア中心の間のクラッド領域ではそれ以外のクラッド領域と比べて電界の強度が高い状態が保たれている。図 3.15 からシングルモードファイバと比べてコアへの電界閉じ込め率が高いことが確認できる。またシングルモードファイバでは正規化周波数 V が大きいという事は動作波長 λ がコア幅 D と比べて相対的に小さいことを示している。短い波長ほど導波モードの条件を満たしやすく正規化伝搬定数 P も大きくなる [7]。このことを微細型 3 コアファイバについて考えると微細型 3 コアファイバではコア間の結合によりシングルモードファイバと比べて実質的なコア径（コア幅）が大きくなると考えられる。そのために同一波長（同一の正規化周波数）においてより相対的に大きなコア幅を持つことになり導波モードを満たしやすく正規化伝搬定数 P がおおきくなったと考える。図 3.15 でも V の増加に伴いシングルモードファイバの特性に漸近していくことが確認された。

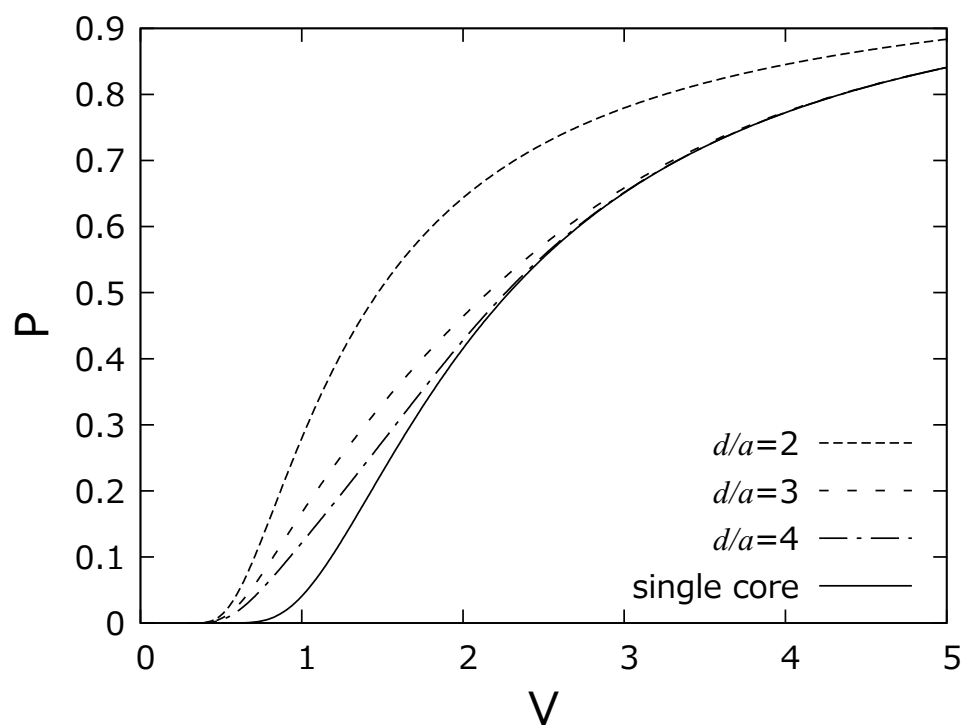


図 3.13: モード $1e^+$ の PV 特性

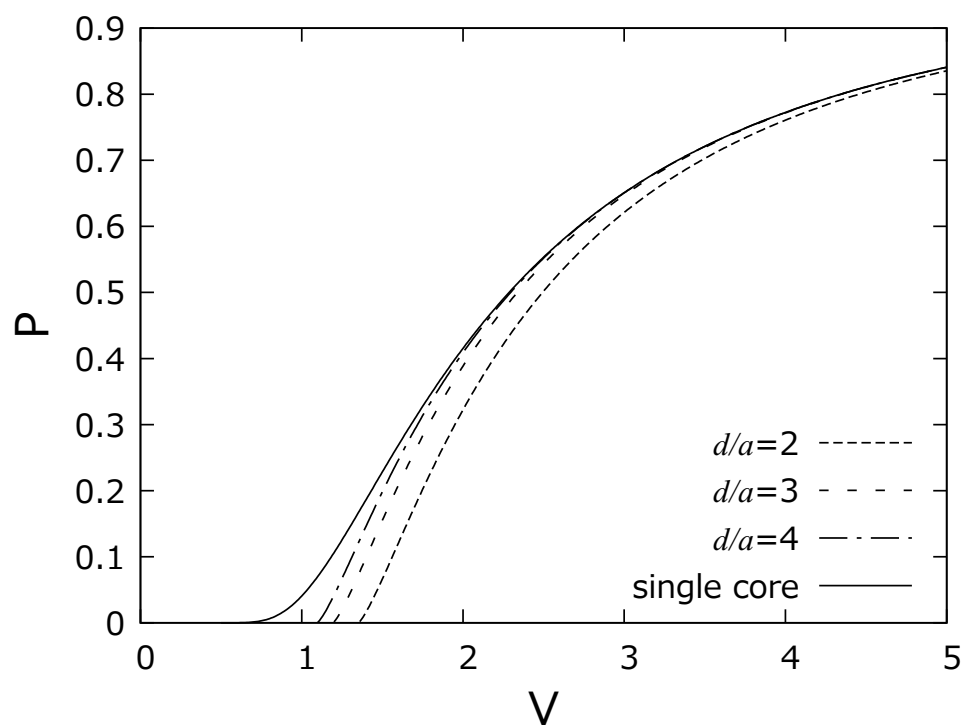


図 3.14: モード $1e^-$ の PV 特性

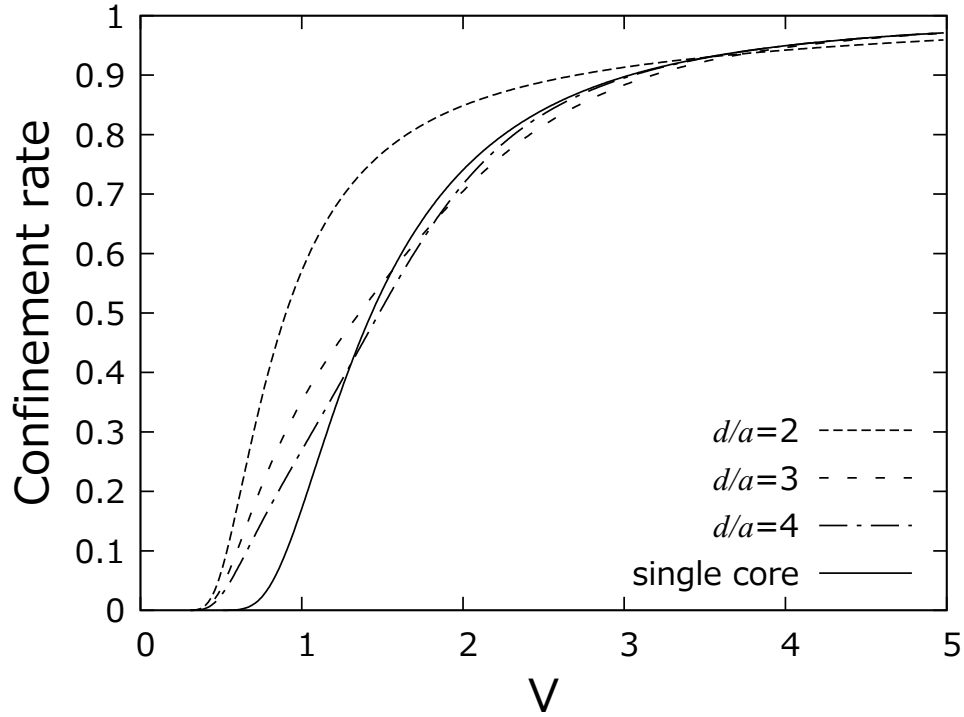


図 3.15: コアへの電界の閉じ込め率

3.4 シングルモードファイバの電磁界との比較結果

図 3.16 にモード $1e^+$ についてパラメータが $d/a = 2 \sim 4$, $\Delta = 0.003$ で条件 (a) シングルモードファイバと微細型 3 コアファイバをコア中心を合わせた状態での横方向電界の NMSE の計算結果を示す. 図 3.16 では $d/a = 3, 4$ では正規化周波数 V が 1.2 辺りまで NMSE は増加しその後減少している. これは V が小さい場合にはシングルモードファイバ, 微細型 3 コアファイバともに正規化伝搬定数 P も小さくなり電磁界はコア, クラッドに平面波的に一様に広がることになる. そのためコア, クラッドの境界はあいまいになるため NMSE は小さくなると考えられる. V の増加に伴い電界のコアへの閉じ込め率は増加し平面波的ではなくなるため NMSE は増加する. しかし V の増加, 閉じ込め率の増加に伴いコア間の結合は強くなるため一つ一つのコアの電界は独立しシングルモードファイバの電界に似ることとなる. そのため NMSE は減少したと考えられる. $d/a = 2$ ではコアが隣接しているため閉じ込め率が増加してもコア間の結合が強く NMSE が小さくならないと考えられる.

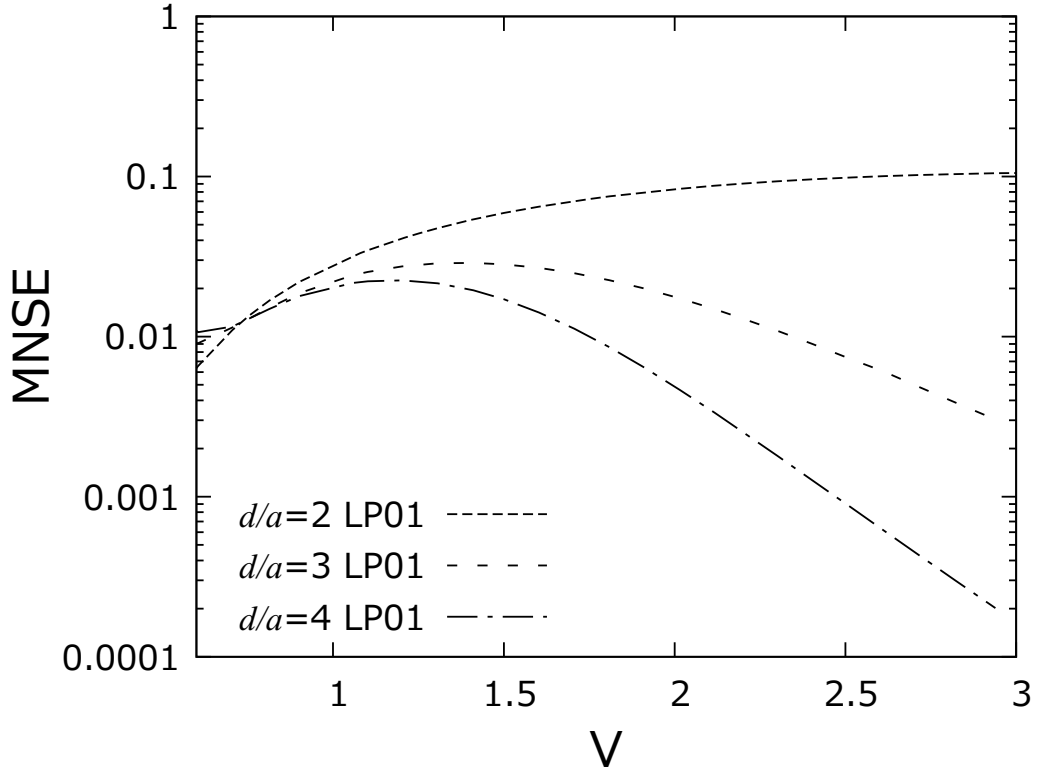


図 3.16: モード $1e^+$ の一つのコアと単一コアの解成分の誤差

図 3.17 にモード $1e^-$ についてパラメータが $d/a = 2 \sim 4\Delta = 0.003$ で条件 (a) で同様に NMSE を計算した結果を示す. 図 3.17 では NMSE の計算結果は直線で減少しており, 横軸は log スケールなので指数関数的に減少していることが確認された. LP01 では各コアの電磁界は同一の直線偏波を保有しているため NMSE の主要因は強度分布によるものと考えられる. LP11 では異なり直線偏波となっておらず, NMSE の要因は強度分布とシングルモードファイバの基本モードからの角度のずれとなる. V が小さく電界が平面波的に一様に広がったとしても角度のずれが存在するため図 3.16 にあるような小さい V での NMSE の減少がないと考えられる. 図 3.18 にモード $1e^+$ で条件 (b) SMF と微細型 3 コアファイバのファイバの中心を合わせ, NMSE が最少となる MFD を計算した結果を示す. 図 3.18 では各 d/a でコア半径で MFD を正規化している. そのため d/a が大きいほど三つのコアの外接円も大きくなる. また d/a が大きいほど閉じ込め率は低くなるため電界はクラッドに広がる. そのためコア半径が等しい場合 d/a が大きいほど MFD の値は大きくなると考えられる. V の増加に伴い閉じ込め率は d/a に関係なく高くなるため V が増加すると MFD は減少する.

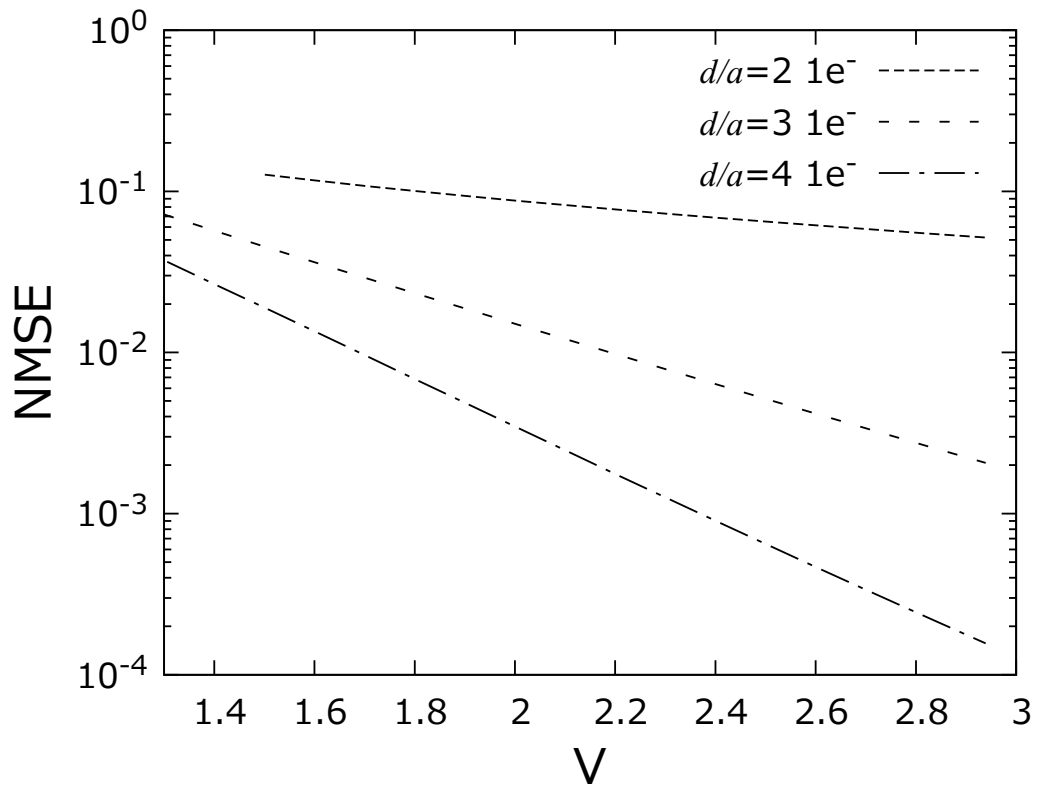


図 3.17: モード $1e^-$ の一つのコアと単一コアの解成分の誤差

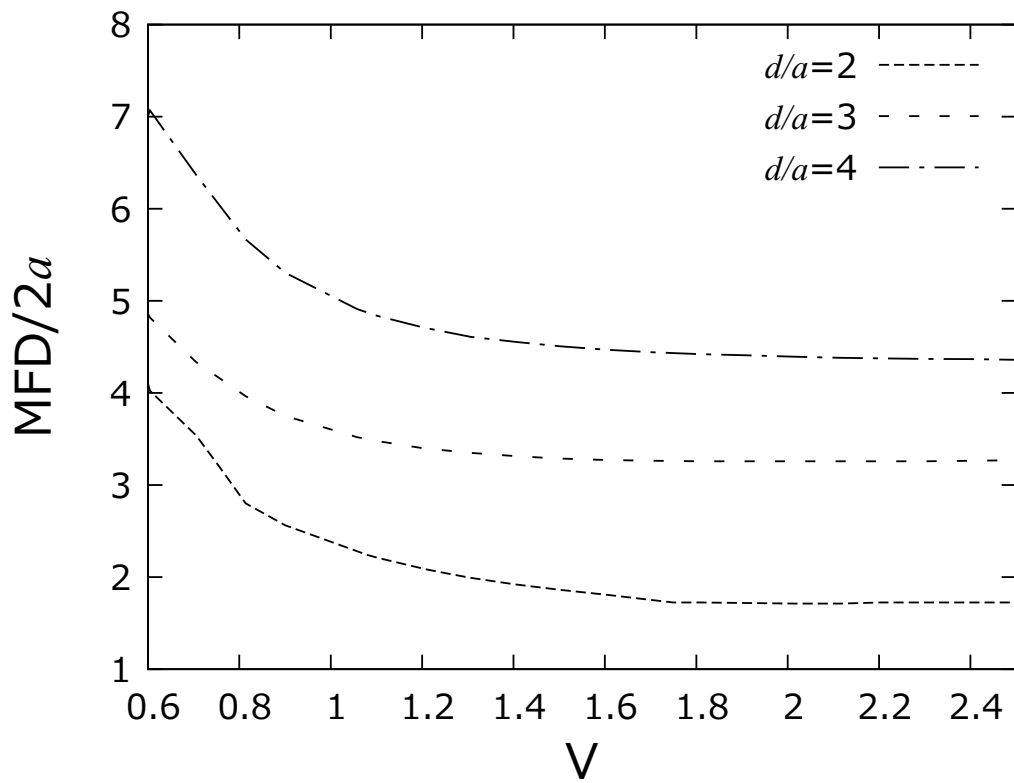


図 3.18: モード $1e^+$ の全体と単一コアの解成分の誤差 (コア直径 $2a$ で正規化)

また図 3.13, 3.15 で示されたように d/a が大きいほど PV 特性はシングルモードファイバに漸近し, 閉じ込め率は小さくなるため d/a が大きいほど小さい場合と比べて電界はクラッドに広がることとなる. 図 3.19 に $d/a = 2, V = 3.0$ での電界強度分布, 図 3.20 に $d/a = 3, V = 3.0$ での強度分布を示す. $V = 3$ なため高い閉じ込め率であるが図 3.19 では 3 つのコアに囲まれたクラッドの部分にも電界が強くあり, 電界強度の最大値がコアの中心ではなくファイバの中心によっているのが確認できる. $d/a = 2$ では V が高い状態でもコア間結合が強く実質的に一つのコアとしてみなせると考えられる.

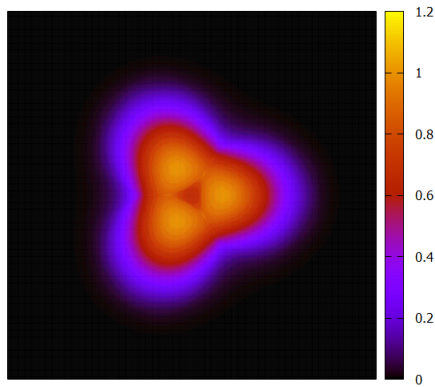


図 3.19: $d/a = 2, V = 3.0$ の横方向電界強度分布

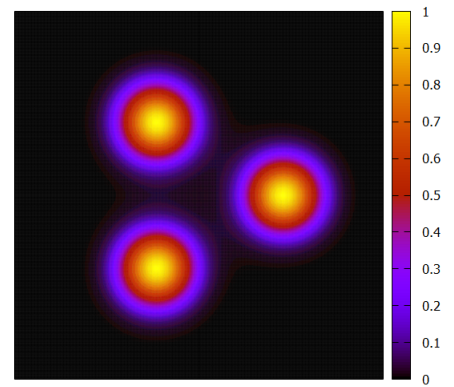


図 3.20: $d/a = 3, V = 3.0$ の横方向電界強度分布

第4章 結論

本研究では微細型3コアファイバについて群論適用点整合法を用いてベクトルモード解析の検証，解析を行った．

検証として境界条件の正規化平均二乗誤差を計算した．その結果として構造によらず高い精度で境界条件が満たされており，群論適用点整合法がベクトルモード解析，伝搬定数の解析に適していることが明らかになった．またコア間隔を広げることによるコア間の結合が弱くなり精度がよくなることも明らかになった．同様に伝搬定数 P が小さく電磁界のコアへの閉じ込めが弱い場合と， P が大きく電磁界の閉じ込めが強い場合についても精度がよくなることが明らかになった．シングルモードファイバとの比較についてはシングルモードファイバとの正規化平均二乗誤差の計算，モードフィールド径の計算を行った．その結果 P が小さい場合と， P が大きい場合ではシングルモードファイバと誤差が小さい電界分布になることが明らかになった． P が小さい場合には電界が平面波的に広がるため実質的に一つのコアとみなせ， P の増加と共にコア間の結合は弱くなることが明らかになった．

謝辞

本研究を進めるにあたり，このような研究の機会を与えていただき，懇切なる指導と助言を賜りました電気通信大学情報理工学研究科情報・ネットワーク工学専攻 來住直人教授に深く感謝を申し上げます。

最後に，多くの知識や助言をいただいた同研究室の先輩，同輩の皆様に感謝いたします。

参考文献

- [1] 「令和二年度版 情報通信白書」,総務省,2020
- [2] 大越 孝敬, 岡本 勝則, 保立 和夫,「光ファイバ」, オーム社, 1985
- [3] E. Yamashita et al, "Modal analysis method for optical fibers with symmetrically distributed multiple cores" Journal of Lightwave technology, Vol. LT-3,NO.2, April 1985.
- [4] 來住 直人, 山下 榮吉, "電磁波問題における群論の応用", 信学技報, MW89-99, 1989.
- [5] 藤永茂, 成田進,「科学や物理のためのやさしい群論入門」, 岩波書店, 2001.
- [6] 斎藤 法久,「3 コアファイバの導波路分散の特性解析」, 電気通信大学 情報・通信工学科 情報通信システムコース 卒業論文, 2016
- [7] 左貝 潤一,「導波光学」, 共立出版,2004
- [8] William H. Press,「Numerical Recipes in C 日本語版」, 技術評論社,1993
- [9] Dietrich Marcuse,"Theory of dielectric optical waveguides",Academic press,1974

発表実績

小林 豪太, 來住 直人, 「微細型 3 コア光ファイバの伝搬特性」, 光ファイバ応用技術研究会, OFT2019-16, 2019

來住 直人, 小林 豪太, 「円周上対称配置 n コア光ファイバのベクトルモード電磁界解析の有効性検証」, OFT2020-51, 2021

小林 豪太, 來住 直人, 「円周上対称配置 3 コア光ファイバのベクトルモード電磁界解析」, OFT2020-61, 2021

付録A 電磁界計算のための手順式

z 軸方向電磁界，横方向電界の計算手順を示す．図 4.1 に示すように微細型 3 コアファイバのファイバの中心を原点とする極座標 (r_c, θ_c) をとり，各コアを原点とする極座標 (r_k, θ_k) ($k = 1, 2, 3$) をとる．

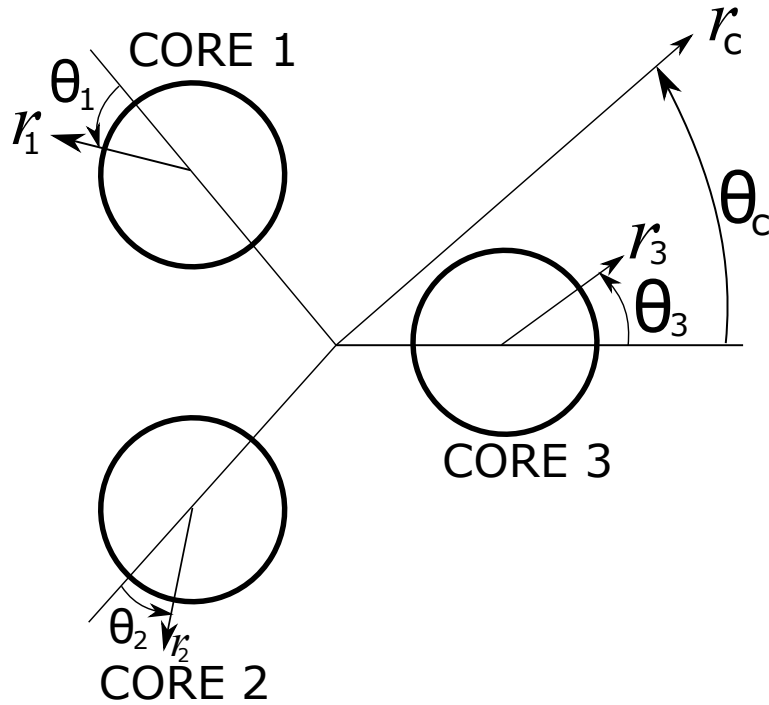


図 4.1: 微細型 3 コアファイバの横断面図とその座標系

光速度を c とし，真空中の誘電率，透磁率を ϵ_0, μ_0 ，コア，クラッドの誘電率を ϵ_1, ϵ_2 とする．微細型 3 コアファイバのパラメータについてコア半径 a ，コア中心間距離 d ，ファイバの中心とコアの中心の距離を R ，比屈折率差 Δ ，コア，クラッド屈折率を n_i ($i = 1, 2$)，動作波長 λ ，モード伝搬定数 l とする，波数 k ，角周波数 ω ，伝搬定数 β 正規化周波数 V は次の式で表される．

$$k = \frac{2\pi}{\lambda} \quad (4.1)$$

$$\omega = kc \quad (4.2)$$

$$\beta = k\sqrt{(n_1^2 - n_2^2)P + n_2^2} \quad (4.3)$$

$$V = kan_1\sqrt{2\Delta} \quad (4.4)$$

V を決めたのちにモード伝搬定数 l を決めると対応する正規化周波数 P が計算可能となる．図 4.1 に点 (r_c, θ_c) をとるとコア ν の中心を原点とする極座標は

$$r_\nu = \sqrt{(r_c \cos(\theta_c) - R \cos(\theta_\nu))^2 + (r_c \sin(\theta_c) - R \sin(\theta_\nu))^2} \quad (4.5)$$

$$\theta_\nu = \arctan \frac{r_c \cos(\theta_c) - R \cos(\theta_\nu)}{r_c \sin(\theta_c) - R \sin(\theta_\nu)} + \pi - \theta_\nu \quad (4.6)$$

$$\theta_1 = 2\pi/3, \theta_2 = 4\pi/3, \theta_3 = 0 \quad (4.7)$$

のように表現される．各コアに対応する極座標を用いて電磁界を計算する．

群論適用点整合法では微細型 3 コアファイバの z 軸方向電磁界は次のように表現される．式中 e, o は偶関数，奇関数を表す．

$$\psi_e^l = \frac{2}{3} \sum_{\nu=1}^3 \left[\cos\left(\frac{2\pi l \nu}{3}\right) \phi_\nu^o - \sin\left(\frac{2\pi l \nu}{3}\right) \phi_\nu^e \right] \quad (4.8a)$$

$$\psi_o^l = \frac{2}{3} \sum_{\nu=1}^3 \left[\sin\left(\frac{2\pi l \nu}{3}\right) \phi_\nu^o + \cos\left(\frac{2\pi l \nu}{3}\right) \phi_\nu^e \right] \quad (4.8b)$$

実際の計算ではどちらかを電界とし，片方を磁界とする．式 (4.5) の ϕ_ν^e, ϕ_ν^o は次のように展開計数を含んだ式で表現される．

ν 番目のコア領域の電界は

$$\phi_\nu^e = \sum_{m=0}^M a_{em} J_m(hr_\nu) \cos(m\theta_\nu) \quad (4.9a)$$

$$\phi_\nu^o = \sum_{m=0}^M a_{om} J_m(hr_\nu) \sin(m\theta_\nu) \quad (4.9b)$$

ν 番目のコア領域の磁界は

$$\phi_\nu^e = \sum_{m=0}^M b_{em} J_m(hr_\nu) \cos(m\theta_\nu) \quad (4.10a)$$

$$\phi_\nu^o = \sum_{m=0}^M b_{om} J_m(hr_\nu) \sin(m\theta_\nu) \quad (4.10b)$$

クラッド領域での電界は

$$\phi_\nu^e = \sum_{m=0}^M c_{em} K_m(pr_\nu) \cos(m\theta_\nu) \quad (4.11a)$$

$$\phi_\nu^o = \sum_{m=0}^M c_{om} K_m(pr_\nu) \sin(m\theta_\nu) \quad (4.11b)$$

クラッド領域での磁界は

$$\phi_\nu^e = \sum_{m=0}^M d_{em} K_m(pr_\nu) \cos(m\theta_\nu) \quad (4.12a)$$

$$\phi_\nu^o = \sum_{m=0}^M d_{om} K_m(pr_\nu) \sin(m\theta_\nu) \quad (4.12b)$$

$$h = \sqrt{\omega^2 \mu_0 \varepsilon_1 - \beta^2} \quad (4.13)$$

$$p = \sqrt{\beta^2 - \omega^2 \mu_0 \varepsilon_2} \quad (4.14)$$

V, P から展開計数がシミュレータで求められる． z 軸方向電磁界を用いて横方向電磁界 $E_r, E_\theta, H_r, H_\theta$ は次のように表現される

$$E_r = -\frac{j}{k^2 n_i^2 - \beta^2} \left(\beta \frac{\partial E_z}{\partial r} + \omega \mu_0 \frac{1}{r} \frac{\partial H_z}{\partial \theta} \right) \quad (4.15a)$$

$$E_\theta = -\frac{j}{k^2 n_i^2 - \beta^2} \left(\beta \frac{1}{r} \frac{\partial E_z}{\partial \theta} - \omega \mu_0 \frac{\partial H_z}{\partial r} \right) \quad (4.15b)$$

$$H_r = -\frac{j}{k^2 n_i^2 - \beta^2} \left(\beta \frac{\partial H_z}{\partial r} - \omega \varepsilon_i \frac{1}{r} \frac{\partial E_z}{\partial \theta} \right) \quad (4.16a)$$

$$H_\theta = -\frac{j}{k^2 n_i^2 - \beta^2} \left(\beta \frac{1}{r} \frac{\partial H_z}{\partial \theta} + \omega \varepsilon_i \frac{\partial E_z}{\partial r} \right) \quad (4.16b)$$

横方向電界を計算する際に z 軸方向の電磁界の偏微分，つまりベッセル関数の偏微分を求める必要があり，その際には第一種ベッセル関数，第二種変形ベッセル関数に関する漸化式を用いる．

$$J'_n = \frac{J_{n-1} - J_{n+1}}{2} \quad (4.17)$$

$$K'_n = \frac{K_{n+1} + K_{n-1}}{-2} \quad (4.18)$$

$$J_{-n} = (-1)^n J_n \quad (4.19)$$

$$K_{-n} = K_n \quad (4.20)$$

$$\frac{1}{u} J_n(u) = \frac{J_{n+1}(u) + J_{n-1}(u)}{2n} \quad (4.21)$$

付録B z軸方向電磁界，横方向電界計算プログラムリスト

プログラムリストでは参考文献 [8] のベッセル関数のプログラムを使用している

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <string.h>
5
6 //ベッセル関数用定数群
7 #define shikii 1e-6
8 #define BIGNO 1.0e10
9 #define BIGNI 1.0e-10
10
11 //ベッセル関数
12 float bessj0(float x);
13 float bessj1(float x);
14 float bessj(int n, float x);
15 float bessi0(float x);
16 float bessk0(float x);
17 float bessi1(float x);
18 float bessk1(float x);
19 float bessk(int n, float x);
20
21 //ベッセル関数使用用
22 double bessjn(int n, double x);
23 double besskn(int n, double x);
24 double bessjd(int n, double x); //微分用
25 double besskd(int n, double x);
26
27 void vector_transe(double x, double y); //横方向電磁界の計算関数
28 void boundary_z(double kakudo3); //z軸方向電磁界の境界条件確認関数
29
30 double Ae[20], Be[20], Ce[20], De[20];
31 double Ao[20], Bo[20], Co[20], Do[20];
32 double h, p;
33 double a, da, k;
34 double Ex, Ey, Er, Es;
35 double m0=1.256637*1e-6;
36 double e0=8.85418782*1e-12;
37 double omega, beta;
38 int number=0;
```

```

39 int option;
40 double R;
41 double corebeta,cladbeta;
42 double delta,n1,n2;
43 double e1,e2;
44 double coreEz,cladEz,coreHz,cladHz;
45 double coreEs,coreEr,cladEs,cladEr,coreHs,coreHr,cladHs,cladHr;
46 int pointnum=0;
47
48 int main(int argc,char *argv[]){
49     FILE *datafile;
50
51     double a1,a2,a3,a4,a5,a6,a7,a8;
52     int i,l,j;
53     double f1,f2;
54     double c=299792458;
55     double lambda;
56     double V1,P1;
57     double r,kakudo;
58     double x,y;
59
60     if(argc==1){
61         printf("1:datafile\n");
62         return 0;
63     }else if(argc != 2){
64         printf("error_%d\n",argc);
65         return 0;
66     }
67
68     //ファイル読み込み
69     datafile=fopen(argv[1],"r");
70     // coffie=fopen(argv[2],"r");
71     if(datafile ==NULL){
72         printf("can_not_open_file\n");
73         return -1;
74     }
75
76     i=0;
77
78     fscanf(datafile,"%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf\n",&a,&delta,&lambda,&da,&k,&omega,&n1,&V1,&P1,&
        beta,&h,&p,&n1,&n2,&e1,&e2);
79
80     while( fscanf(datafile, "%lf_%lf_%lf_%lf_%lf_%lf_%lf_%lf",&a1,&a2,&
        a3,&a4,&a5,&a6,&a7,&a8) !=EOF){
81         Ae[i]=a1;
82         Ao[i]=a2;
83         Bo[i]=a3;
84         Be[i]=a4;

```



```

85     Ce[i]=a5;
86     Co[i]=a6;
87     Do[i]=a7;
88     De[i]=a8;
89     number++;
90     i++;
91 }
92
93 R=da*a/sqrt(3);
94 corebeta=pow(k*n1,2)-pow(beta,2);
95 cladbeta=pow(k*n2,2)-pow(beta,2);
96
97 double x,y;
98 x=0;
99 y=0;
100
101 vector_transe(x,y);
102 printf("%e_%e_%e_%e\n",x,y,Ez,Ey);
103
104 fclose(datafile);
105 return 0;
106 }
107
108 void vector_transe(double x,double y){
109
110     double r,kakudo;
111     r=sqrt(x*x+y*y);
112     kakudo=atan2(y,x);
113
114     Er=0;
115     Es=0;
116     Ex=0;
117     Ey=0;
118
119     double n=3;
120     double l=1;
121     double v;
122     int i;
123
124     double r1,r2,r3;
125     double kakudo1,kakudo2,kakudo3;
126     double kakudo1o,kakudo2o,kakudo3o; Ezscore=0,Hzrcore=0,Ezsclad=0,
        Hzrclad=0,Ezrcore=0,Ezrclad=0,Hzsclad=0;
127
128     r1=sqrt(pow(r*cos(kakudo)-R*cos(2*M_PI/3.0),2)+pow(r*sin(kakudo)-R*
        sin(2*M_PI/3.0),2));
129     r2=sqrt(pow(r*cos(kakudo)-R*cos(4*M_PI/3.0),2)+pow(r*sin(kakudo)-R*
        sin(4*M_PI/3.0),2));
130     r3=sqrt(pow(r*cos(kakudo)-R*cos(0),2)+pow(r*sin(kakudo)-R*sin

```

```

        (0),2));
131
132 kakudo1=atan2(r*sin(kakudo)-R*sin(2*M_PI/3.0),r*cos(kakudo)-R*cos(2*
    M_PI/3.0));
133 kakudo2=atan2(r*sin(kakudo)-R*sin(4*M_PI/3.0),r*cos(kakudo)-R*cos(4*
    M_PI/3.0));
134 kakudo3=atan2(r*sin(kakudo)-R*sin(0),r*cos(kakudo)-R*cos(0));
135
136 kakudo1o=kakudo1;
137 kakudo2o=kakudo2;
138 kakudo3o=kakudo3;
139
140 kakudo1=kakudo1+4*M_PI/3.0;
141 kakudo2=kakudo2+2*M_PI/3.0;
142 kakudo3=kakudo3;
143
144 double temp1,temp2;
145 double temp3,temp4;
146 temp3=0;
147 temp4=0;
148
149 if(r1<a){
150     v=1;
151     temp1=0;
152     temp2=0;
153     for(i=0;i<number;i++){
154         temp1 +=-Ae[i]*bessjn(i,h*r1)*sin(i*kakudo1)*i;
155     }
156
157     for(i=0;i<number;i++){
158         temp2 +=Ao[i]*bessjn(i,h*r1)*cos(i*kakudo1)*i;
159     }
160     Ezscore =2/n*(cos(2*M_PI*v*1/n)*temp1-sin(2*M_PI*v*1/n)*temp2);
161
162     temp1=0;
163     temp2=0;
164     for(i=0;i<number;i++){
165         temp1 +=Ae[i]*bessjd(i,h*r1)*h*cos(i*kakudo1);
166     }
167
168     for(i=0;i<number;i++){
169         temp2 +=Ao[i]*bessjd(i,h*r1)*h*sin(i*kakudo1);
170     }
171
172     Ezrcore =2/n*(cos(2*M_PI*v*1/n)*temp1-sin(2*M_PI*v*1/n)*temp2);
173
174     temp1=0;
175     temp2=0;
176     for(i=0;i<number;i++){

```

```

177     temp1 +=Be[i]*bessjd(i,h*r1)*h*cos(i*kakudo1);
178 }
179
180 for(i=0;i<number;i++){
181     temp2+=Bo[i]*bessjd(i,h*r1)*h*sin(i*kakudo1);
182 }
183
184 Hzrcore=2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
185
186 temp1=0;
187 temp2=0;
188 for(i=0;i<number;i++){
189     temp1 +=-Be[i]*bessjn(i,h*r1)*sin(i*kakudo1)*i;
190 }
191
192 for(i=0;i<number;i++){
193     temp2 +=Bo[i]*bessjn(i,h*r1)*cos(i*kakudo1)*i;
194 }
195
196 Hzscore=2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
197
198 temp1=0;
199 temp2=0;
200
201 Es=beta/corebeta*(Ezscore/r1-k*n2/beta*Hzrcore);
202 Er=beta/corebeta*(Ezrcore+k*n2/(beta*r1)*Hzscore);
203
204 Ex=cos(kakudo1o)*Er+cos(kakudo1o+M_PI/2.0)*Es; //Ex
205 Ey=sin(kakudo1o)*Er+sin(kakudo1o+M_PI/2.0)*Es; //Ey
206 }else if(r2<a){
207
208     v=2;
209     temp1=0;
210     temp2=0;
211     for(i=0;i<number;i++){
212         temp1 +=-Ae[i]*bessjn(i,h*r2)*sin(i*kakudo2)*i;
213     }
214
215     for(i=0;i<number;i++){
216         temp2 +=Ao[i]*bessjn(i,h*r2)*cos(i*kakudo2)*i;
217     }
218     Ezscore =2/n*(cos(2*M_PI*v*l/n)*temp1-sin(2*M_PI*v*l/n)*temp2);
219
220     temp1=0;
221     temp2=0;
222     for(i=0;i<number;i++){
223         temp1 +=Ae[i]*bessjd(i,h*r2)*h*cos(i*kakudo2);
224     }
225

```

```

226     for(i=0;i<number;i++){
227         temp2 +=Ao[i]*bessjd(i,h*r2)*h*sin(i*kakudo2);
228     }
229
230     Ezrcore =2/n*(cos(2*M_PI*v*l/n)*temp1-sin(2*M_PI*v*l/n)*temp2);
231
232     temp1=0;
233     temp2=0;
234     for(i=0;i<number;i++){
235         temp1 +=Be[i]*bessjd(i,h*r2)*h*cos(i*kakudo2);
236     }
237
238     for(i=0;i<number;i++){
239         temp2+=Bo[i]*bessjd(i,h*r2)*h*sin(i*kakudo2);
240     }
241
242     Hzrcore=2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
243
244     temp1=0;
245     temp2=0;
246     for(i=0;i<number;i++){
247         temp1 +=-Be[i]*bessjn(i,h*r2)*sin(i*kakudo2)*i;
248     }
249
250     for(i=0;i<number;i++){
251         temp2 +=Bo[i]*bessjn(i,h*r2)*cos(i*kakudo2)*i;
252     }
253
254     Hzscore=2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
255
256     temp1=0;
257     temp2=0;
258
259     Es=beta/corebeta*(Ezscore/r2-k*n2/beta*Hzrcore);
260     Er=beta/corebeta*(Ezrcore+k*n2/(beta*r2)*Hzscore);
261
262     Ex=cos(kakudo2o)*Er+cos(kakudo2o+M_PI/2.0)*Es; //Ex
263     Ey=sin(kakudo2o)*Er+sin(kakudo2o+M_PI/2.0)*Es; //Ey
264 }else if(r3<a){
265
266     v=3;
267     temp1=0;
268     temp2=0;
269     for(i=0;i<number;i++){
270         temp1 +=-Ae[i]*bessjn(i,h*r3)*sin(i*kakudo3)*i;
271     }
272
273     for(i=0;i<number;i++){
274         temp2 +=Ao[i]*bessjn(i,h*r3)*cos(i*kakudo3)*i;

```

```

275     }
276     Ezscore =2/n*cos(2*M_PI*v*l/n)*temp1;
277
278     temp1=0;
279     temp2=0;
280     for(i=0;i<number;i++){
281         temp1 +=Ae[i]*bessjd(i,h*r3)*h*cos(i*kakudo3);
282     }
283
284     for(i=0;i<number;i++){
285         temp2 +=Ao[i]*bessjd(i,h*r3)*h*sin(i*kakudo3);
286     }
287
288     Ezrcore =2/n*cos(2*M_PI*v*l/n)*temp1;
289
290
291     temp1=0;
292     temp2=0;
293     for(i=0;i<number;i++){
294         temp1 +=Be[i]*bessjd(i,h*r3)*h*cos(i*kakudo3);
295     }
296
297     for(i=0;i<number;i++){
298         temp2+=Bo[i]*bessjd(i,h*r3)*h*sin(i*kakudo3);
299     }
300
301     Hzrcore=2/n*cos(2*M_PI*v*l/n)*temp2;
302
303     temp1=0;
304     temp2=0;
305     for(i=0;i<number;i++){
306         temp1 +=-Be[i]*bessjn(i,h*r3)*sin(i*kakudo3)*i;
307     }
308
309     for(i=0;i<number;i++){
310         temp2 +=Bo[i]*bessjn(i,h*r3)*cos(i*kakudo3)*i;
311     }
312
313     Hzscore=2/n*cos(2*M_PI*v*l/n)*temp2;
314
315     Es=beta/corebeta*(Ezscore/r3-k*n2/beta*Hzrcore);
316     Er=beta/corebeta*(Ezrcore+k*n2/(beta*r3)*Hzscore);
317
318     Ex=cos(kakudo3o)*Er+cos(kakudo3o+M_PI/2.0)*Es; //Ex
319     Ey=sin(kakudo3o)*Er+sin(kakudo3o+M_PI/2.0)*Es; //Ey
320 }else{
321
322     v=1;
323     temp1=0;

```

```

324     temp2=0;
325
326     for(i=0;i<number;i++){
327         temp1 +=-Ce[i]*besskn(i,p*r1)*sin(i*kakudo1)*i;
328     }
329     for(i=0;i<number;i++){
330         temp2 +=Co[i]*besskn(i,p*r1)*cos(i*kakudo1)*i;
331     }
332
333     Ezsclad=2/n*(cos(2*M_PI*v*l/n)*temp1-sin(2*M_PI*v*l/n)*temp2);
334
335     temp1=0;
336     temp2=0;
337
338     for(i=0;i<number;i++){
339         temp1 +=Ce[i]*besskd(i,p*r1)*p*cos(i*kakudo1);
340     }
341
342     for(i=0;i<number;i++){
343         temp2 +=Co[i]*besskd(i,p*r1)*p*sin(i*kakudo1);
344     }
345
346     Ezrclad =2/n*(cos(2*M_PI*v*l/n)*temp1-sin(2*M_PI*v*l/n)*temp2);
347
348     temp1=0;
349     temp2=0;
350
351     for(i=0;i<number;i++){
352         temp1 +=De[i]*besskd(i,p*r1)*p*cos(i*kakudo1);
353     }
354
355     for(i=0;i<number;i++){
356         temp2 +=Do[i]*besskd(i,p*r1)*p*sin(i*kakudo1);
357     }
358
359     Hzsclad =2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
360
361     temp1=0;
362     temp2=0;
363
364     for(i=0;i<number;i++){
365         temp1 +=-De[i]*besskn(i,p*r1)*sin(i*kakudo1)*i;
366     }
367
368     for(i=0;i<number;i++){
369         temp2 +=Do[i]*besskn(i,p*r1)*cos(i*kakudo1)*i;
370     }
371
372     Hzsclad=2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);

```

```

373
374     temp1=beta/cladbeta*(Ezscld/r1-k*n2/beta*Hzrcld); //Es
375     temp2=beta/cladbeta*(Ezrcld+k*n2/(beta*r1)*Hzscld); //Er
376
377     temp3=cos(kakudo1o)*temp2+cos(kakudo1o+M_PI/2.0)*temp1; //Ex
378     temp4=sin(kakudo1o)*temp2+sin(kakudo1o+M_PI/2.0)*temp1; //Ey
379
380     Er+=temp3*cos(kakudo3)+temp4*cos(M_PI/2.0-kakudo3);
381     Es+=-temp3*sin(kakudo3)+temp4*sin(M_PI/2.0-kakudo3);
382
383     v=2;
384     temp1=0;
385     temp2=0;
386
387     for(i=0;i<number;i++){
388         temp1 +=-Ce[i]*besskn(i,p*r2)*sin(i*kakudo2)*i;
389     }
390     for(i=0;i<number;i++){
391         temp2 +=Co[i]*besskn(i,p*r2)*cos(i*kakudo2)*i;
392     }
393
394     Ezscld=2/n*(cos(2*M_PI*v*l/n)*temp1-sin(2*M_PI*v*l/n)*temp2);
395
396     temp1=0;
397     temp2=0;
398
399     for(i=0;i<number;i++){
400         temp1 +=Ce[i]*besskd(i,p*r2)*p*cos(i*kakudo2);
401     }
402
403     for(i=0;i<number;i++){
404         temp2 +=Co[i]*besskd(i,p*r2)*p*sin(i*kakudo2);
405     }
406
407     Ezrcld =2/n*(cos(2*M_PI*v*l/n)*temp1-sin(2*M_PI*v*l/n)*temp2);
408
409     temp1=0;
410     temp2=0;
411
412     for(i=0;i<number;i++){
413         temp1 +=De[i]*besskd(i,p*r2)*p*cos(i*kakudo2);
414     }
415
416     for(i=0;i<number;i++){
417         temp2 +=Do[i]*besskd(i,p*r2)*p*sin(i*kakudo2);
418     }
419
420     Hzrcld =2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
421

```

```

422     temp1=0;
423     temp2=0;
424
425     for(i=0;i<number;i++){
426         temp1 +=-De[i]*besskn(i,p*r2)*sin(i*kakudo2)*i;
427
428     }
429     for(i=0;i<number;i++){
430         temp2 +=Do[i]*besskn(i,p*r2)*cos(i*kakudo2)*i;
431     }
432
433     Hzsclad=2/n*(sin(2*M_PI*v*1/n)*temp1+cos(2*M_PI*v*1/n)*temp2);
434
435     temp1=beta/cladbeta*(Ezsclad/r2-k*n2/beta*Hzrclad); //Es
436     temp2=beta/cladbeta*(Ezrclad+k*n2/(beta*r2)*Hzsclad); //Er
437
438     temp3=cos(kakudo2o)*temp2+cos(kakudo2o+M_PI/2.0)*temp1; //Ex
439     temp4=sin(kakudo2o)*temp2+sin(kakudo2o+M_PI/2.0)*temp1; //Ey
440
441     Er+=temp3*cos(kakudo3)+temp4*cos(M_PI/2.0-kakudo3);
442     Es+=-temp3*sin(kakudo3)+temp4*sin(M_PI/2.0-kakudo3);
443
444     v=3;
445     temp1=0;
446     temp2=0;
447
448     for(i=0;i<number;i++){
449         temp1 +=-Ce[i]*besskn(i,p*r3)*sin(i*kakudo3)*i;
450     }
451     for(i=0;i<number;i++){
452         temp2 +=Co[i]*besskn(i,p*r3)*cos(i*kakudo3)*i;
453     }
454
455     Ezsclad=2/n*(cos(2*M_PI*v*1/n)*temp1-sin(2*M_PI*v*1/n)*temp2);
456
457     temp1=0;
458     temp2=0;
459
460     for(i=0;i<number;i++){
461         temp1 +=Ce[i]*besskd(i,p*r3)*p*cos(i*kakudo3);
462     }
463
464     for(i=0;i<number;i++){
465         temp2 +=Co[i]*besskd(i,p*r3)*p*sin(i*kakudo3);
466     }
467
468     Ezrclad =2/n*(cos(2*M_PI*v*1/n)*temp1-sin(2*M_PI*v*1/n)*temp2);
469
470

```



```

471     temp1=0;
472     temp2=0;
473
474     for(i=0;i<number;i++){
475         temp1 +=De[i]*besskd(i,p*r3)*p*cos(i*kakudo3);
476     }
477
478     for(i=0;i<number;i++){
479         temp2 +=Do[i]*besskd(i,p*r3)*p*sin(i*kakudo3);
480     }
481
482     Hzrclad =2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
483
484     temp1=0;
485     temp2=0;
486
487     for(i=0;i<number;i++){
488         temp1 +=-De[i]*besskn(i,p*r3)*sin(i*kakudo3)*i;
489     }
490
491     for(i=0;i<number;i++){
492         temp2 +=Do[i]*besskn(i,p*r3)*cos(i*kakudo3)*i;
493     }
494
495     Hzsclad=2/n*(sin(2*M_PI*v*l/n)*temp1+cos(2*M_PI*v*l/n)*temp2);
496
497     Es+=beta/cladbeta*(Ezscld/r3-k*n2/beta*Hzrclad); //Es
498     Er+=beta/cladbeta*(Ezrclad+k*n2/(beta*r3)*Hzsclad); //Er
499
500     Ex=cos(kakudo3o)*Er+cos(kakudo3o+M_PI/2.0)*Es; //Ex
501     Ey=sin(kakudo3o)*Er+sin(kakudo3o+M_PI/2.0)*Es; //Ey
502
503 }
504 }
505
506
507 void boundary_z(double kakudo3){
508     double r=sqrt(pow(R+a*cos(kakudo3),2)+pow(a*sin(kakudo3),2));
509     double kakudo=atan2(a*sin(kakudo3),R+a*cos(kakudo3));
510     int i;
511     double n=3;
512     double l=1;
513     double v;
514     Ex=0;
515     Ey=0;
516     Er=0;
517     Es=0;
518
519     double ans;

```

```

520 double r1,r2,r3;
521 double kakudo1,kakudo2;
522 double kakudo1o,kakudo2o,kakudo3o;
523
524 r1=sqrt(pow(r*cos(kakudo)-R*cos(2*M_PI/3.0),2)+pow(r*sin(kakudo)-R*
    sin(2*M_PI/3.0),2));
525 r2=sqrt(pow(r*cos(kakudo)-R*cos(4*M_PI/3.0),2)+pow(r*sin(kakudo)-R*
    sin(4*M_PI/3.0),2));
526 r3=sqrt(pow(r*cos(kakudo)-R*cos(0),2)+pow(r*sin(kakudo)-R*sin
    (0),2));
527
528
529 kakudo1=atan2(r*sin(kakudo)-R*sin(2*M_PI/3.0),r*cos(kakudo)-R*cos(2*
    M_PI/3.0));
530 kakudo2=atan2(r*sin(kakudo)-R*sin(4*M_PI/3.0),r*cos(kakudo)-R*cos(4*
    M_PI/3.0));
531
532 kakudo1o=kakudo1;
533 kakudo2o=kakudo2;
534 kakudo3o=kakudo3;
535
536 kakudo1=kakudo1+4*M_PI/3.0;
537 kakudo2=kakudo2+2*M_PI/3.0;
538
539 double tmp1,tmp2,tmp3;
540
541 coreEz=0;
542 v=3;
543 tmp1=0;
544 tmp2=0;
545
546 for(i=0;i<number;i++){
547     tmp1 +=Ae[i]*bessjn(i,h*a)*cos(i*kakudo3);
548 }
549 for(i=0;i<number;i++){
550     tmp2 +=Ao[i]*bessjn(i,h*a)*sin(i*kakudo3);
551 }
552
553 coreEz=2.0/n*(cos(2*M_PI*1*v/n)*tmp1-sin(2*M_PI*1*v/n)*tmp2);
554
555
556 cladEz=0;
557 v=1;
558 tmp1=0;
559 tmp2=0;
560
561 for(i=0;i<number;i++){
562     tmp1 +=Ce[i]*besskn(i,p*r1)*cos(i*kakudo1);
563 }

```

```

564     for(i=0;i<number;i++){
565         tmp2 +=Co[i]*besskn(i,p*r1)*sin(i*kakudo1);
566     }
567
568     cladEz +=2.0/n*(cos(2*M_PI*1*v/n)*tmp1-sin(2*M_PI*1*v/n)*tmp2);
569
570     v=2;
571     tmp1=0;
572     tmp2=0;
573
574     for(i=0;i<number;i++){
575         tmp1 +=Ce[i]*besskn(i,p*r2)*cos(i*kakudo2);
576     }
577     for(i=0;i<number;i++){
578         tmp2 +=Co[i]*besskn(i,p*r2)*sin(i*kakudo2);
579     }
580
581     cladEz +=2.0/n*(cos(2*M_PI*1*v/n)*tmp1-sin(2*M_PI*1*v/n)*tmp2);
582
583     v=3;
584     tmp1=0;
585     tmp2=0;
586     for(i=0;i<number;i++){
587         tmp1 +=Ce[i]*besskn(i,p*a)*cos(i*kakudo3);
588     }
589
590     for(i=0;i<number;i++){
591         tmp2 +=Co[i]*besskn(i,p*a)*sin(i*kakudo3);
592     }
593
594     cladEz +=2.0/n*(cos(2*M_PI*1*v/n)*tmp1-sin(2*M_PI*1*v/n)*tmp2);
595
596     coreHz=0;
597     cladHz=0;
598     v=3;
599     tmp1=0;
600     tmp2=0;
601
602     for(i=0;i<number;i++){
603         tmp1 +=Be[i]*bessjn(i,h*a)*cos(i*kakudo3);
604     }
605     for(i=0;i<number;i++){
606         tmp2 +=Bo[i]*bessjn(i,h*a)*sin(i*kakudo3);
607     }
608     coreHz=2.0/n*(sin(2*M_PI*1*v/n)*tmp1+cos(2*M_PI*1*v/n)*tmp2);
609
610     v=1;
611     tmp1=0;
612     tmp2=0;

```

```

613     for(i=0;i<number;i++){
614         tmp1 +=De[i]*besskn(i,p*r1)*cos(i*kakudo1);
615     }
616
617     for(i=0;i<number;i++){
618         tmp2 +=Do[i]*besskn(i,p*r1)*sin(i*kakudo1);
619     }
620
621     cladHz +=2.0/n*(sin(2*M_PI*1*v/n)*tmp1+cos(2*M_PI*1*v/n)*tmp2);
622
623     v=2;
624     tmp1=0;
625     tmp2=0;
626     for(i=0;i<number;i++){
627         tmp1 +=De[i]*besskn(i,p*r2)*cos(i*kakudo2);
628     }
629
630     for(i=0;i<number;i++){
631         tmp2 +=Do[i]*besskn(i,p*r2)*sin(i*kakudo2);
632     }
633
634     cladHz +=2.0/n*(sin(2*M_PI*1*v/n)*tmp1+cos(2*M_PI*1*v/n)*tmp2);
635
636     v=3;
637     tmp1=0;
638     tmp2=0;
639     for(i=0;i<number;i++){
640         tmp1 +=De[i]*besskn(i,p*a)*cos(i*kakudo3);
641     }
642
643     for(i=0;i<number;i++){
644         tmp2 +=Do[i]*besskn(i,p*a)*sin(i*kakudo3);
645     }
646
647     cladHz +=2.0/n*(sin(2*M_PI*1*v/n)*tmp1+cos(2*M_PI*1*v/n)*tmp2);
648 }
649
650 float bessj0(float x){
651     float ax,z;
652     double xx,y,ans,ans1,ans2;
653
654     if((ax=fabs(x))<8.0){
655         y=x*x;
656         ans1=57568490574.0+y*(-13362590354.0+y*(651619640.7
657             +y*(-11214424.18+y*(77392.33017+y*(-184.9052456)))));
658         ans2=57568490411.0+y*(1029532985.0+y*(9494680.718
659             +y*(59272.64853+y*(267.8532712+y*1.0))));
660         ans=ans1/ans2;
661     }else {

```

```

662     z=8.0/ax;
663     y=z*z;
664     xx=ax-0.785398164;
665     ans1=1.0+y*(-0.1098628627e-2+y*(0.2734510407e-4
666         +y*(-0.2073370639e-5+y*0.2093887211e-6)));
667     ans2 = -0.1562499995e-1+y*(0.1430488765e-3
668         +y*(-0.6911147651e-5+y*(0.7621095161e-6
669         -y*0.934935152e-7)));
670     ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-z*sin(xx)*ans2);
671 }
672 return ans;
673 }
674
675 float bessj1(float x){
676     float ax,z;
677     double xx,y,ans,ans1,ans2;
678
679     if((ax=fabs(x))<8.0){
680         y=x*x;
681         ans1=x*(72362614232.0+y*(-7895059235.0+y*(242396853.1
682             +y*(-2972611.439+y*(15704.48260+y*(-30.16036606))))));
683         ans2=144725228442.0+y*(2300535178.0+y*(18583304.74
684             +y*(99447.43394+y*(376.9991397+y*1.0))));
685         ans=ans1/ans2;
686     } else {
687         z=8.0/ax;
688         y=z*z;
689         xx=ax-2.356194491;
690         ans1=1.0+y*(0.183105e-2+y*(-0.3516396496e-4
691             +y*(0.2457520174e-5+y*(-0.240337019e-6)));
692         ans2=0.04687499995+y*(-0.2002690873e-3
693             +y*(0.8449199096e-5+y*(-0.88228987e-6
694             +y*0.105787412e-6)));
695         ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-z*sin(xx)*ans2);
696         if(x<0.0) ans = -ans;
697     }
698     return ans;
699 }
700
701 float bessj(int n,float x){
702     float bessj0(float x);
703     float bessj1(float x);
704     int j,jsum,m;
705     float ax,bj,bjm,bjp,sum,tox,ans;
706     float ACC=40;
707     ax=fabs(x);
708     if(ax == 0.0)
709         return 0.0;
710     else if (ax > (float) n) {

```

```

711     tox=2.0/ax;
712     bjm=bessj0(ax);
713     bj=bessj1(ax);
714     for (j=1;j<n;j++){
715         bjp=j*tox*bj-bjm;
716         bjm=bj;
717         bj=bjp;
718     }
719     ans=bj;
720 } else {
721     tox=2.0/ax;
722     m=2*((n+(int) sqrt(ACC*n))/2);
723     jsum=0;
724     bjp=ans=sum=0.0;
725     bj=1.0;
726     for(j=m;j>0;j--){
727
728         bjm=j*tox*bj-bjp;
729         bjp=bj;
730         bj=bjm;
731
732
733         if(fabs(bj)>BIGNO){
734             bj*=BIGNI;
735             bjp*=BIGNI;
736             ans*=BIGNI;
737             sum*=BIGNI;
738         }
739         if(jsum) sum += bj;
740         jsum =!jsum;
741         if(j==n) ans=bjp;
742     }
743     sum=2.0*sum-bj;
744     ans/=sum;
745 }
746 return x< 0.0 && (n & 1) ? -ans : ans;
747 }
748
749 float bessj0(float x){
750     double y;
751     float ax,ans;
752     if((ax=fabs(x)) <3.75){
753         y=x/3.75;
754         y *=y;
755         ans=1.0+y*(3.5156229+y*(3.0899424+y*(1.2067494
756             +y*(0.2659732+y*(0.360768e-1+y*(0.45813e-2))))));
757     } else {
758         y=3.75/ax;
759         ans=(exp(ax)/sqrt(ax))*(0.39894228+y*(0.1328592e-1

```

```

760         +y*(0.225319e-2+y*(-0.157565e-2+y*(0.916281e-2
761         +y*(-0.2057706e-1+y*(0.2635537e-1+y*(-0.1647633e-1
762         +y*0.392377e-2))))))));
763     }
764     return ans;
765 }
766
767 float bessk0(float x){
768     //K0 をかえす
769     float bessj0(float x);
770     double y,ans;
771
772     if(x <=2.0){
773         y=x*x/4.0;
774         ans=(-log(x/2.0)*bessj0(x))+(-0.57721566+y*(0.42278420
775         +y*(0.23069756+y*(0.3488590e-1+y*(0.262698e-2
776         +y*(0.10750e-3+y*0.74e-5))))));
777     } else {
778         y=2.0/x;
779         ans=(exp(-x)/sqrt(x))*(1.25331414+y*(-0.7832358e-1
780         +y*(0.2189568e-1+y*(-0.1062446e-1+y*(0.587872e-2
781         +y*(-0.251540e-2+y*0.53208e-3))))));
782     }
783     return ans;
784 }
785
786 float bessj1(float x){
787     float ax,ans;
788     double y;
789     if((ax=fabs(x)) <3.75){
790         y=x/3.75;
791         y*=y;
792         ans=ax*(0.5+y*(0.87890594+y*(0.51498869+y*(0.15084934
793         +y*(0.2658733e-1+y*(0.301532e-2+y*0.32411e-3))))));
794     } else {
795         y=3.75/ax;
796         ans=0.2282967e-1+y*(-0.2895312e-1+y*(0.1787654e-1
797         -y*0.420059e-2));
798         ans=0.39894228+y*(-0.3988024e-1+y*(-0.362018e-2
799         +y*(0.163801e-2+y*(-0.1031555e-1+y*ans))));
800         ans*=(exp(ax)/sqrt(ax));
801     }
802     return x < 0.0 ? -ans : ans;
803 }
804
805 float bessk1(float x){
806     float bessj1(float x);
807     double y,ans;
808     if(x<=2.0){

```

```

809     y=x*x/4.0;
810     ans=(log(x/2.0)*bessi1(x))+(1.0/x)*(1.0+y*(0.15443144
811         +y*(-0.67278579+y*(-0.18156897+y*(-0.1919402e-1
812         +y*(-0.110404e-2+y*(-0.4686e-4))))));
813 } else {
814     y=2.0/x;
815     ans=(exp(-x)/sqrt(x))*(1.25331414+y*(0.23498619
816         +y*(-0.3655620e-1+y*(0.1504268e-1+y*(-0.780353e-2
817         +y*(0.325614e-2+y*(-0.68245e-3))))));
818 }
819 return ans;
820 }
821
822 float bessk(int n,float x){
823     float bessk0(float x);
824     float bessk1(float x);
825     int j;
826     float bk,bkm,bkp,tox;
827
828     tox=2.0/x;
829     bkm=bessk0(x);
830     bk=bessk1(x);
831     for(j=1;j<n;j++){
832         bkp=bkm+j*tox*bk;
833         bkm=bk;
834         bk=bkp;
835     }
836     return bk;
837 }
838
839 double bessjn(int n,double x){
840     double ans;
841     if(n==0){
842         ans=bessj0(x);
843     }else if(n==1){
844         ans=bessj1(x);
845     }else if(n==2){
846         ans=bessj(2,x);
847     }else{
848         ans=bessj(n,x);
849     }
850     return ans;
851 }
852
853 double besskn(int n,double x){
854     double ans;
855     if(n==0){
856         ans=bessk0(x);
857     }else if(n==1){

```



```

858     ans=bessk1(x);
859 }else if(n==2){
860     ans=bessk(2,x);
861 }else{
862     ans=bessk(n,x);
863 }
864 return ans;
865 }
866 double bessjd(int n,double x){
867     double ans;
868     if(n==0){
869         ans=-bessjn(1,x);
870     }else if(n==1){
871         ans=(bessjn(0,x)-bessjn(2,x))/2.0;
872     }else{
873         ans=(bessjn(n-1,x)-bessjn(n+1,x))/2.0;
874     }
875     return ans;
876 }
877 double besskd(int n,double x){
878     double ans;
879     if(n==0){
880         ans=-besskn(1,x);
881     }else{
882         ans=(besskn(n-1,x)+besskn(n+1,x))/(-2.0);
883     }
884 }
885 return ans;
886 }

```
